CHANNEL ESTIMATION OF SINGLE –TASK AND MULTI –TASK NETWORKS BASED ON TRANSFORM DOMAIN ANALYSIS

BY

ALI MOHAMMED AL-MOHAMMEDI

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

TELECOMMUNICATIONS ENGINEERING

DECEMBER 2017

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **ALI MOHAMMED AL-MOHAMMEDI** under the direction of his thesis adviser and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN TELECOMMUNICATIONS ENGINEERING**.

Thesis Committee

Dr. Mohamed Deriche (Adviser)

Azzedine Zerguine

Dr. Azzedine Zerguine (Member)

Dr. Samir Al-Ghadban (Member)

Dr. Ali Ahmad Al-Shaikhi
Department Chairman

Dr. Salam A. Zummo
Dean of Graduate Studies

16/5/14

Date

Dedication to My Parents:
Mr. Mohammed Ahmed Hadi Al-Mohammedi
and
Mrs. Mahiah Abdullah Jafeir Al-Janahi

ACKNOWLEDGMENTS

I would like to express my gratitude to the members of my thesis committee members, namely: Dr. Mohamed Deriche (Advisor), Dr. Azzedine Zerguine, and Dr. Samir Al-Ghadbanm for their guidance and support. It was my good luck to work with them and learn from the diversified expertise of Dr. Mohamed Deriche in detection and estimation and Dr. Azzedine Zerguine in adaptive filtering, and Dr. Samir Al-Ghadbanm in simulation of communication systems (Equalizer) under the topic of "Channel Estimation of Single-Task and Multi-Task Networks Based On Transform Domain Analysis".

I am extremely indebted to my thesis academic advisor, Dr. Mohamed Deriche, for his guidance, cooperation, and the time he provided generously in mentoring me over the past five years as part of my research to complete my thesis in the digital signal processing research laboratory.

It is my pleasure to acknowledge the support and cooperation of Dr. Muhammad Omer Bin Saeed (National University of Sciences and Technology, NUST, Pakistan) in this thesis.

It is my pleasure to acknowledge the support and facilities provided by the digital signal processing research laboratory at King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Kingdom of Saudi Arabia.

I would also like to acknowledge the support and high level of education provided by KFUPM throughout my master's degree in telecommunications engineering. Moreover, I would like to acknowledge the support and funds provided by King Abdulaziz for Science and Technology (KACST), Riyadh, Kingdom of Saudi Arabia, under the research grant AT-37-1758.

My father "Mr. Mohammed Ahmed Hadi Al-Mohammedi",my mother "Mahiah Abdullah Al-Janahi", my grandmother "Mrs. Fatmah Abdullah Zamman" (passed away during last submission of thesis), my sisters, my brothers "Engr. Essam Mohammed Al-Mohammedi" and "Mr. Ahmed Mohammed Almohammedi", and my uncles "specially, Mr. Amin Ahmed Hadi Al-Mohammedi" have always been a great source of inspiration for my educational endeavors. My sincere thanks are also due to other relatives, to name only a few: Engr. Fouad Al-Saleh, Mr. Tareq Al-Ansari, Mr. Abdulrahman Al-Mohash, Mr. Nabeel Al-Atiq.

Ali Mohammed Ahmed Almohammedi

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Symbols

x	letter in normal font denotes a vector or a scalar
X	capital letter in normal font denotes a matrix
$\mathbf{x}$	boldface letter denotes a random scalar
$\mathbf{X}$	boldface capital letter denotes a random matrix
$J(i)$	cost function at iteration i
$e(i)$	output estimation error at iteration i
$d(i)$	desirable signal at iteration i
$y(i)$	output signal at iteration i
w^o	optimum filter (a column vector)
w_i	weight estimate at iteration i (a column vector)
$\tilde{w}_i$	weight error vector at iteration i (a column vector)
u_i	input signal or regressor at iteration i (a row vector)
$u_i^{'}$	input signal or regressor at iteration i (matrix)
$v(i)$	additive white gaussian noise
μ	constant step-size
$\mu(i)$	variable step-size at iteration i
σ_i^2	power estimate at iteration i (a column vector)
δ	small constant
β	convergence factor
γ	convergence factor
α	exponential weighting parameter
I_N	identity matrix of size $N \times N$
$\boldsymbol{\psi}^i$	global weight-update (adaptive step) at iteration i (a column vector)
$\tilde{\boldsymbol{\psi}}^i$	global weight error at iteration i (a column vector)
$\boldsymbol{\phi}^{i-1}$	global weight-update (combine step) at iteration $i-1$ (a column vector)
$\boldsymbol{U}_i$	global block regression matrix at iteration i
$\boldsymbol{d}_i$	global desired vector at iteration i (a column vector)
$\boldsymbol{v}_i$	global background noise vector at iteration i (a column vector)
D	constant global step-size matrix
R_u	autocorrelation matrix of regressor

Abbreviations

LMS	Least Mean Square
NLMS	Normalized Least Mean Square
RLS	Recursive Least Square
DLMS	Diffusion Least Mean Square
MSE	Mean Square Error
MSD	Mean Square Deviation
EMSE	Excess Mean Square Error
MMSE	Minimum Mean Square Deviation
IEEE	Institute of Electrical and Electronics Engineering
WSN	Wireless Sensor Network
PAN	Personal Area Network
LAN	Local Area Network
WLAN	Wireless Local Area Network
TDLMS	Transform Domain Least Mean Square
DCTLMS	Discrete Cosine Transform Least Mean Square
TDVSS	Transform Domain Variable Step-Size Least Mean Square
NVSTDLMS	New Variable Step-Size Transform Domain Least Mean Square
VSSTDLMS	Variable Step-Size Transform Domain Least Mean Square
VSSLMS	Variable Step-Size Least Mean Square
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DWT	Discrete Wavelet Transform
ATC	Adapt-Then-Combine
CTA	Combine-Then-Adapt
ISI	Inter-Symbol Interference
PDF	Probability Density Function

Operators

$E\{\mathbf{x}\}$	expected value of random variable $\mathbf{x}$
$col\{a, b\}$	column vector with entries a and b
$diag\{a\}$	diagonal matrix with entries read from the column a
$.^T$	matrix transportation
$.^*$	complex conjugation of scalars or Hermitian transposition for matrices
$A \odot B$	block Kronecker product of two block matrices A and B
$a \otimes B$	product of each element of a matrix B by a scalar a
$\| x \|_W^2$	x^*Wx for a column vector x and positive-definite matrix W

THESIS ABSTRACT

NAME: Ali Mohammed Al-Mohammedi

TITLE OF STUDY: Channel Estimation of Single-Task and Multi-Task Networks Based on Transform Domain Analysis

MAJOR FIELD: Telecommunications Engineering

DATE OF DEGREE: December 2017

In this thesis, new variable step-size transform domain (VSSTD) algorithms are developed for Diffusion Least Mean-Square (DLMS) single-task and multi-task networks with system identification application over Wireless Sensor Networks (WSNs).

Our main contributions are the theoretical derivations of convergence analysis of numerous variants of VSSTDLMS algorithms. Another important contribution is the introduction of adaptive combiners for the case of multi-task networks.

In this work, unlike the work carried out in the literature, the technique of transform domain is employed to reduce the eigenvalue spread of the input regressor autocorrelation, hence the correlation among input regressors is minimized. Our simulations showed improvement performance compared to the traditional DLMS.

THESIS ABSTRACT (ARABIC)

ملخص الرسالة

الأسم: علي بن محمد بن احمد المحمدي

عنوان الرسالة: تقدير قنوات الشبكات موحدة-المهام ومتعددة-المهام معتمدة على تحليل مجال التحويل

الدرجة: ماجستير في العلوم

التخصص: هندسة اتصالات

تاريخ التخرج: ديسمبر 2017م

في هذه الأطروحة، فإن خوارزميات جديدة في مجال التحويل متغير الحجم (VSSTD) تم تطويره لمربع المتوسط المنتشر (DLMS) لشبكات موحدة-المهام ومتعددة-المهام مع تطبيق تعريف النظام عبر شبكات الأستشعار اللاسلكية (WSNs).

مساهماتنا الرئيسية هي الأشتقاقات النظرية لتحليل التقارب لأعداد متنوعة من الخوارزميات VSSTDLMS. ومن المساهمات المهمة الأخرى تقديم مدمجات تكيفية لحالة الشبكات متعددة-المهام.

في هذا العمل، على عكس العمل المنجز في الأدب، يتم توظيف تقنية مجال التحويل لتقليل انتشار القيم الذاتية للارتباط الذاتي لمصفوفات الأشاره المدخله، وبالتالي تقليل الأرتباط بين الاشارات المدخله. أظهرت عمليات المحاكاة لدينا تحسن في الأداء بالمقارنة مع DLMS التقليدي.

CHAPTER 1

INTRODUCTION

1.1 Background

Channel Estimation is an essential process for modeling communication channels and analyzing their effects. Adaptive filters play an essential role in channel estimation research as well as in noise estimation in relation to equalizers.

A common index for measuring the quality of an estimator is the Mean-Square Deviation (MSD). Many types of adaptive filters have been proposed for channel estimation. Among them is the famous Least Mean Square (LMS) filter, which is referred to as single-node LMS network in this thesis. It processes the instantaneous input and the desired data to produce online estimation of the channel. In contrast, the traditional Wiener Filter (WF) is considered as an off-line estimator according to the Minimum Mean Square Error (MMSE) criterion and is relies on the autocorrelation of the input signal and cross correlation of the input and the desired signal [1].

Note however that the input and the desired data which propagate through the communication channel might be correlated which can degrade the performance of the estimator. The high spread of the eigenvalues of the input autocorrelation matrix is an indication of such correlation. This encouraged us to discuss the problem of channel estimation in the transform domain to mitigate the correlation issue.

Wireless Sensor Networks (WSN) are an important part of modern Local Area Networks (LAN) commonly known as IEEE 802. The standard for Wireless Local Area Networks (WLAN) is known as IEEE 802.11. WSNs are built based on the Personal Area Networks (PAN) IEEE 802.15 (IEEE 802.15.1 to IEEE 802.15.7) which are small scale networks. In this thesis, we discuss the problem of the Diffusion LMS (DLMS) algorithm over wireless networks in transform domain.

The DLMS WSN can be classified into three main types as shown in Figures 1.1, 1.2, and 1.3 namely [2]:

1. Single-Task Networks: The entire network topology has to identify the same parameter vector. This topology will be used throughout this text.

2. Multi-Task Networks: Each agent attempts to estimate its own parameter vector.

3. Clustered Multi-Task Networks: Nodes are gathered into several clusters where each individual cluster shares the same parameter vector. This topology will be used in this text. We will also refer to these as multi-task networks for simplicity.

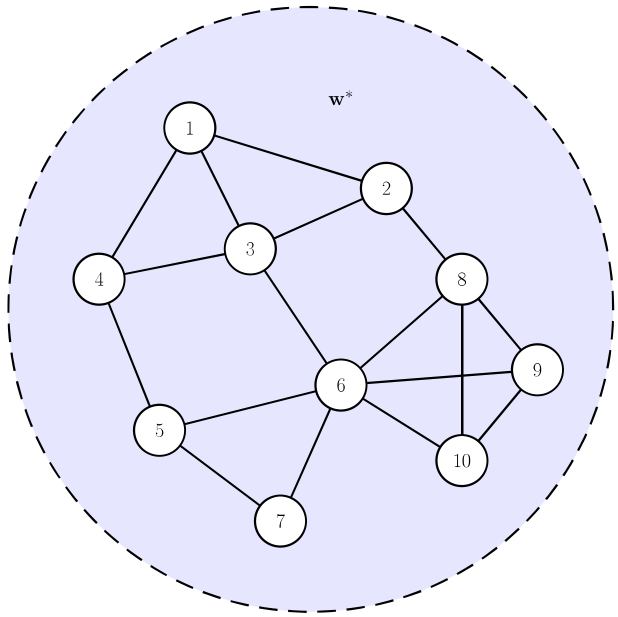

Figure 1.1: Single-Task Networks

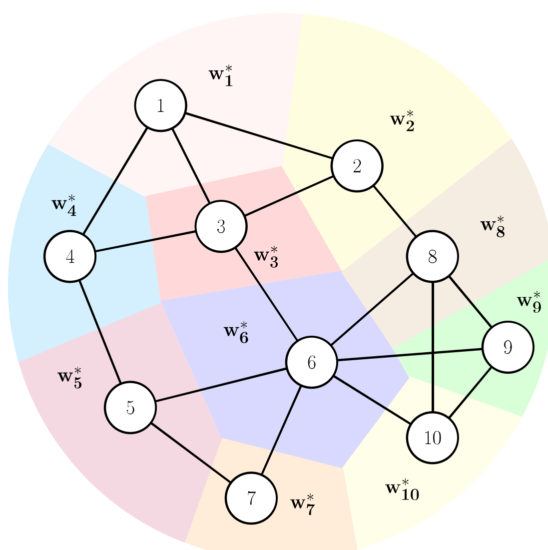

Figure 1.2: Multi-Task Networks

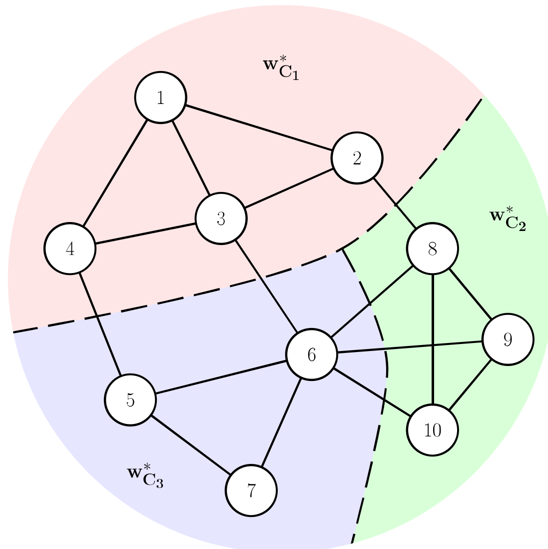

Figure 1.3: Clustered Multi-Task Networks

1.2 Problem Statement

Traditional equalizers perform poorly and have higher learning curves when using conventional adaptive filters such as LMS and RLS. Although conventional adaptive filters can estimate the communication channel and obtain its weight coefficients, but the performance is not as expected (higher MSD). Even worse, if the input signal (input data) to the communication channel is correlated, the performance of the adaptive filter gets degraded and slows down its convergence. A state of the art estimator that can achieve a good learning curve for unknown communication channels is obtained by reformulating the LMS in the transform domain (TDLMS). This reduces the eigenvalue spread of the autocorrelation of the received data, hence improving the performance of the estimator. Conventional LMS can be enhanced further by considering variable step-size LMS (VSSLMS) which leads to a better learning curve. In addition to channel equalization, channel estimation can improve the learning curve and the performance of the filter. The poor performance of multi-task networks degraded clearly when the network topology is unknown apriori or keeps changing over learning iterations. This is true due to different estimation interests among individual clusters, since each cluster has its own optimum filter. The static combiner fails when it aggregates a mix of clusters without distinguishing between which cluster an agent belongs to. Hence, there is a need to develop an adaptive combiner to provide more weights to agents within a cluster and lower weights to other agents belonging to other clusters.

1.3 Research Objectives

The objective of this thesis is to analyze diffusion filters (DLMS) using variable step-size transform domain algorithms (VSSTDLMS) for single-node networks without collaboration and extend it to single-task networks. In this study, the work of single-task networks will include performance analysis, derivations and simulations for:

1. Traditional Transform Domain DLMS (TDLMS)
2. Discrete Cosine Transform DLMS (DCTLMS)

3. Transform Domain Variable Step-Size DLMS (TDVSS)

4. New Variable Step-Size Transform-Domain DLMS (NVSTDLMS)

5. Optimal Step-Size Transform Domain DLMS (VSSTDLMS)

6. Variable Step-Size DLMS (VSSLMS)

Adaptive combiners will then be developed to convert previous analysis of single-task networks to some cases of multi-task networks.

1.4 Main Contributions

The main contributions of this thesis are:

1. We derived the equations of variable step-size transform domain algorithms for single-task DLMS networks, namely:

   (a) Transform Domain DLMS (TDLMS)

   (b) Discrete Cosine Transform DLMS (DCTLMS)

   (c) Transform Domain Variable Step-Size DLMS (TDVSS)

   (d) New Variable Step-Size Transform-Domain DLMS (NVSTDLMS)

   (e) Optimal Variable Step-Size Transform-Domain DLMS (VSSTDLMS)

   (f) Variable Step-Size DLMS (VSSLMS)

2. We extended the above algorithms to multi-task DLMS networks with adaptive combiners.

1.5 Work Organization

The thesis is organized into five chapters, namely: In Chapter 2, we provide an extensive literature review that discusses existing work. In Chapter 3, we develop new variable step-size transform domain algorithms for single-task networks over WSNs. In Chapter 4, the previous algorithms for Chapter 3 are extended to multi-task networks over WSNs with the assistance of adaptive clustering. Both chapters provide extensive experimental results. In Chapter 5, we summarize our experimental results and conclude the thesis with some recommendations for future work.

CHAPTER 2

RELATED WORKS

Adaptive filters have been used as powerful techniques in the analysis of wideband channels (the communication channel is time-variant) by providing good channel estimators such as the LMS and other techniques. Here, a variant of the LMS algorithm will be discussed in order to enhance performance when there is fading and consequently, the input signal has high autocorrelation. In this thesis, we refer to the LMS technique as single-node LMS network. Variable step-size transform domain least mean square algorithms (VSSTDLMS) will also be discussed based on recent research that considers single-node LMS [3].

The idea of single-node LMS networks has been recently implemented for single-task distributed networks to improve global performance [4]. We will discuss VSSTDLMS algorithms in this thesis and analyze their performance. Finally, DLMS multi-task networks have been recently developed in WSNs, but their performance is poor [2]. We will discuss VSSTDLMS algorithms as well as adaptive combiners to mitigate the issue of poor estimates. For the multi-task networks, we will focus on the most recent techniques using transform domain formulation. In the sequel, we evaluate the quality of the DLMS in order to develop algorithms to improve performance.

2.1 Adaptive Filters

Adaptive filters are typical types of linear filters that are used with optimization criteria. The filter accepts input variable parameters to adjust the parameter of interest at the output. Mainly, there are two types of adaptive filters, namely: Least Mean Square (LMS) and Recursive Least Square (RLS).

The convergence rate of LMS in (Figure 2.1) is governed by the step-size that can be either fixed or variable. In this discussion, we will use the following cost function:

$$J(i) = E\{e^2(i)\} \tag{2.1}$$

where the error $e(i)$, is presented by

$$\begin{aligned} e(i) &= d(i) - y(i) \\ &= u_i w^o - u_i w_i + v(i) \\ &= u_i(w^o - w_i) + v(i) \\ &= u_i \tilde{w}_i + v(i) \end{aligned} \tag{2.2}$$

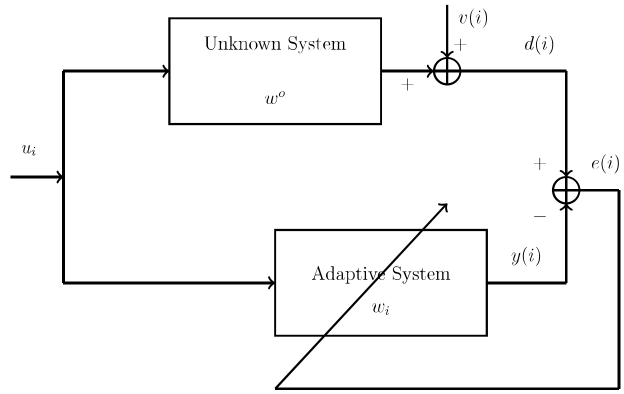

Figure 2.1: Pictorial Illustrative Drawing of LMS Based on System Identification Application.

where $d(i)$ is the desired signal, $y(i)$ is the output signal, w^o is the optimum filter column vector of length $M \times 1$ and M is the length of the filter, u_i is the input signal with length $1 \times M$, $v(i)$ is Additive White Gaussian Noise (AWGN) with zero-mean and variance of σ_v^2, and $\tilde{w}_i$ is the apriori error column vector at iteration i.

The traditional LMS has a constant step-size and the coefficient updates are:

$$w_i = w_{i-1} + \mu u_i e(i) \tag{2.3}$$

where u_i is the transmitted signal in time domain, $e(i) = d(i) - u_i w_i$ is the output error.

Several variant algorithms were originally developed for single node networks to minimize the MSD which improve the performance of the LMS. Survey of these algorithms are given below

1. Traditional Transform Domain LMS (TDLMS) [5–10]

2. Discrete Cosine Transform LMS (DCTLMS) [8]

3. Transform Domain Variable Step-Size LMS (TDVSS) [9]

4. New Variable Step-Size Transform-Domain LMS (NVSTDLMS) [11]

5. Optimal Variable Step-Size Transform Domain LMS (VSSTDLMS) [10]

6. Variable Step-Size LMS (VSSLMS) [3]

The conventional TDLMS algorithm shown in Figure 2.2 also has a constant step-size and the coefficient update equation is defined as [5–10]:

$$w_i = w_{i-1} + \frac{\mu}{\delta + \sigma_i^2} e(i) u_i \tag{2.4}$$

where u_i is the transmitted signal in transformed domain using orthogonal transform matrix such as DFT or DCT, $e(i) = d(i) - w_i u_i$ is the generated error, δ is a narrow constant to avoid infinity solution when the power estimate σ_i^2 is zero. This is also called Normalized LMS (NLMS). The power estimate σ_i^2 is defined as

$$\sigma_i^2 = \beta \sigma_{i-1}^2 + (1 - \beta)|u_i|^2 \tag{2.5}$$

where the convergence factor $0 \ll \beta < 1$.

The DCTLM algorithms [8] can further be adjusted with variable step-size as

$$\mu_i = \beta \mu_{i-1} + \gamma(1 - \beta) \frac{1}{\delta + \frac{1}{M} u_i^{T'} u_i^{'}} \tag{2.6}$$

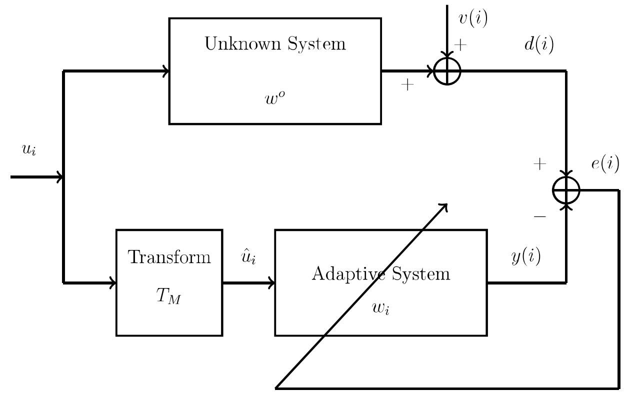

Figure 2.2: Pictorial Illustrative Drawing of System Identification Utilizing Transform Domain LMS.

where $0 \ll \gamma < 1$ is the convergence factor, and $u_i^{'} = col\{u_i, u_{i-1}, ..., u_{i-(L-1)}\}$ is the input signal Matrix of the past L coefficients.

The TDVSS algorithm [9] is a time-varying approach which initially uses a large step-size to improve speed rate of the learning rate then uses a narrow step-size when error becomes negligible or the algorithm is near convergence. The time-varying step-size is given:

$$A(i) = \alpha\mu(i) + \frac{\gamma}{L} \sum_{j=i-(L-1)}^{i} e^2(j) \tag{2.7}$$

$$\mu(i+1) = \begin{cases} A(i), & \text{if } i = kL, \text{and } \mu_{min} < A(i) < \mu_{max} \\ \mu_{max}, & \text{if } i = kL, \text{and } A(i) \geq \mu_{max} \\ \mu_{min}, & \text{if } i = kL, \text{and } A(i) \leq \mu_{min} \\ \mu(i), & \text{if } i \neq kL \end{cases} \tag{2.8}$$

where $0 \ll \alpha < 1$ is the convergence factor, $L \geq 1$ is the time-varying integer number, and $\mu(i)$ is variable step-size.

An optimal approach called VSSTDLMS [10] has been developed with a quick re-learning speed when an unexpected disruption occurs and is given by,

$$\begin{aligned} \rho(i) &= \gamma\rho(i-1) + (1-\gamma)(e^2(i)\hat{u}_i^T\hat{u}_i) \\ \eta(i) &= \gamma\eta(i-1) + (1-\gamma)(e(i)e(i-1)) \\ \mu(i) &= \frac{|\eta(i)|}{\rho(i)} \end{aligned} \tag{2.9}$$

where $\hat{u}(i) = M^{-1}u_i$ with $M^2 = diag[\sigma_1^2, \sigma_2^2, \ldots \sigma_M^2]$.

In another approach which is known as NVSTDLMS [11], the variable step-size is updated based on all output errors available and it also has a very fast re-convergence speed. The NVSTDLMS is represented by

$$\varepsilon(i) = \alpha\varepsilon(i-1) + e^2(i) \tag{2.10}$$

$$\mu(i) = \begin{cases} \frac{\varepsilon_1(i)}{\varepsilon_2(i)}, & \text{if } \mu_{min} < \mu(i) < \mu_{max} \\ \mu_{max}, & \text{if } \mu(i) \geq \mu_{max} \\ \mu_{min}, & \text{if } \mu(i) \leq \mu_{min} \end{cases} \tag{2.11}$$

where $\varepsilon_1(i)$ and $\varepsilon_2(i)$ are step-sizes obtained by considering all output errors available with the exponential weighting parameters $0 < \alpha_1 < \alpha_2 < 1$ where the power estimator was calculated using (2.5).

2.2 Literature Review

In this section, we summarize existing research in three directions of LMS related research: single-node LMS networks, single-task DLMS networks, and multi-task DLMS networks. Since the early nineties, substantial amount of research has been done in variable step-size transform domain algorithms for single-node LMS networks. Single-task DLMS networks have been developed over the last two decades, while work on the variable step-size methodologies has begun in early 2010. Multi-task DLMS networks have been proposed also around 2010s, but they have been shown a very poor performance.

In single-task DLMS networks, time or transform domain variable step-size algorithms were proposed. The famous variable step-size algorithms include DCTLMS [12], NVSTDLMS [13], TDLMS [5–10], TDVSS [9], Optimal Step-Size (VSST-DLMS) [10], and VSSLMS [14]. It was shown that the performance of traditional single-task LMS networks improves substantially with variable step-size algorithms.

The performance of diffusion strategies over single node was also improved by inserting variable step-size algorithms [11] for uncorrelated input regressors while variable step-size transform domain algorithms [5–10] were sought to reduce the correlation in the input signal.

In a different approach to transform domain adaptive filters [15], it was suggested to process the discrete-time signals using block technique of linear convolution or circular convolution which is seen as an approximation. Also, multirate adaptive filters were also proposed where the input signals are transformed by using multirate filters then decimated by a factor that depends on the degree of aliasing.

In single-task networks, three methods of cooperation, namely; Incremental Least Mean Square (ILMS), Diffusion LMS (DLMS), and Poroplastic Diffusion LMS (PDLMS) were initially formulated and their performance was analyzed in WSNs [4, 16]. The diffusion topology was preferred over incremental topology due to its immunity to link failures. These methods estimate a common parameter in a distributed manner by minimizing global cost function (mean square error) which aggregates the estimates from the neighbouring nodes. The work on single-node LMS networks has been extended to WSNs which contain many nodes in order to enhance the performance in terms of mean square deviation (MSD) of single-task parameter estimation [14] using variable step-size technique.

In multi-task networks, diffusion strategies have been successfully introduced and analysed under AWGN which produces poor filter quality when the combiner was selected with respect to the right stochastic rule. This approach has been generalized for any multi-task network with adaptive combiner or clustering resulting in significant performance improvement. The theoretical derivation of MSD performance however has so far not been discussed [17, 18]. In this thesis, we will work on the theoretical derivation of multi-task networks with adaptive combiners.

Table 2.1: Table of Commonly Used References Related To The Thesis

Method	**Algorithm**	**[Ref]**
Variable Step-Size LMS	TDLMS NLMS	[5–11]
	DCTLMS	[8]
	TDVSS	[9]
	VSSTDLMS	[10]
	NVSTDLMS	[11]
	LSVSSLMS	[19]
	Sigmoid	[13, 20]
	Marr	[21]
LMS in Transform Domain	TDLMS (DWT, DFT, DCT)	[5, 22]
	DWTLMS	[23, 24]
Single-Task Networks	Incremental (ILMS)	[16]
	Differential (DLMS)	[4]
	VSS-DLMS	[25]
	NCVSSLMS	[26]
VSSLMS Single-Task	Good Performance	[14, 27]
Multi-Task Networks	Bad Performance	[17]

In this thesis, the aforementioned single-task WSNs [4] will be reformulated with respect to adaptive clustering [17] in order to achieve a promising multi-task WSN. Ultimately, the mean-square deviation or performance will be scaled and improved extremely by considering the variable step-size transform domain algorithms.

A summary of the main research works focusing on adaptive filtering is mentioned in Table 2.1.

2.3 Applications of Adaptive Filters

Before leaving this chapter, we will briefly list the main applications of adaptive filters:

1. System Identification: Here, adaptive filters (e.g. LMS) are used to find the

unknown parameters of a system (called optimum filter). The unknown system receives a data signal and produces a desired signal while the adaptive filter also receives the data signal and generates an output signal. The error between the desired data and the output data is considered to adjust the filter parameters which is called the training stage. This process continues until the error becomes negligible or the estimates converge. In this work, we use the structure of system identification for channel estimation.

2. Noise Cancellation: Echo Cancellation is an example of noise cancellation. It is equipped with hybrid filter to reroute the far end speech to local culler and vice versa. The desired data is mixed with important local culler data and the far end speech.

3. Linear Prediction: Regression is an example of linear prediction. Also, forward linear prediction is helpful when the last data is lost, while backward linear prediction is preferable when the first data is lost.

4. Equalization: Adaptive filters estimate the inverse coefficients of the unknown system. This technique is widely applied to mitigate the Inter-Symbol Interference (ISI) phenomenon in digital communication.

In this thesis, channel estimation is considered as an application of system identification which identifies the parameter estimate of a communication system.

In our work, two methods are discussed for DLMS algorithm:

1. Adapt-Then-Combine (ATC): It preforms the adaptation stage for node k

to estimate the intermediate parameter then it aggregates all local estimates of the neighbours of node k including its own intermediate estimate (called combine stage) to get the local estimate of node k.

2. Combine-Then-Adapt (CTA): It has the same definition of ATC, but in reverse order.

In this work, we consider ATC technique of DLMS to derive the theoretical MSD performance in chapters 3 and 4. On the other hand, we will use the CTA technique of DLMS to simulate all algorithms using Monte Carlo simulations. We show that the theoretical and actual simulations of ATC and CTA have a good agreement with CTA showing marginally better performance than ATC.

CHAPTER 3

TRANSFORM DOMAIN VARIABLE STEP-SIZE DLMS ALGORITHMS (SINGLE-TASK NETWORKS)

The estimation problem of single-task WSNs is initially formulated in order to derive weight error equations. The analysis of the mean transient and the mean square of weight error are very important in evaluating the design of the underlying network. In the sequel, we extend the work in variable step-size transform-domain algorithms in single-task networks (DLMS) to seek a better mean-square deviation or performance of the global network. Finally, intensive Monte-Carlo Simulations are discussed to validate the performance of the derived algorithms.

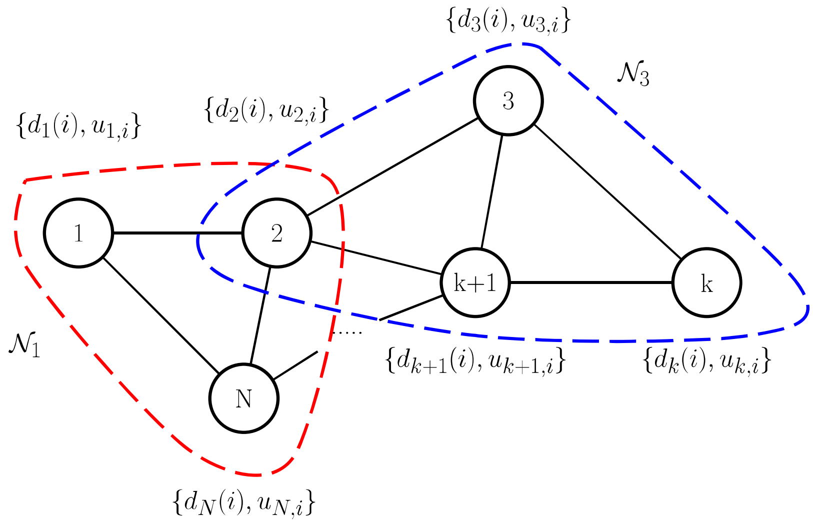

Figure 3.1: Single-Task Networks with N Nodes.

3.1 Problem Formulation

In this problem, we seek the estimates of $M-1$ unknown filter coefficients, w^o, recorded at N distributed sensors nodes as shown in Figure 3.1 [4]. A regressor row vector $u_{k,i}$ with size $1 \times M$ and a scalar reading $d_k(i)$ for node k, where $k = 1, ..., N$, are obtained for every time i and considered as a time realization $\{d_k(i), u_{k,i}\}$ of the random pair $\{\boldsymbol{d}_k, \boldsymbol{u}_k\}$ with zero mean. The global representation is [4]:

$$\boldsymbol{U}_c \triangleq col\{\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_N\} \quad (N \times M) \tag{3.1a}$$

$$\boldsymbol{d} \triangleq col\{\boldsymbol{d}_1, \boldsymbol{d}_2, ..., \boldsymbol{d}_N\} \quad (N \times 1) \tag{3.1b}$$

The block regression matrix is defined by:

$$\boldsymbol{U} \triangleq diag\{\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_N\} \quad (N \times NM) \tag{3.2}$$

Based on the above, we have:

$$\boldsymbol{U}_c = \boldsymbol{U}Q \quad (N \times M) \tag{3.3}$$

where

$$Q = col\{I_M, I_M, ..., I_M\} \quad (NM \times M) \tag{3.4}$$

The Combine-then-Adapt (CTA) technique is represented by:

$$\begin{aligned} \phi_{k,i-1} &= \sum_{l \in \mathcal{N}_k, i-1} c_{kl} \psi_{l,i-1} \\ \psi_{k,i} &= \phi_{k,i-1} + \mu_k u_{k,i}^* (d_k(i) - u_{k,i} \phi_{k,i-1}) \end{aligned} \tag{3.5}$$

where $\phi_{k,i}$ is the combination or aggregation stage, $\psi_{k,i}$ is the adaptation stage, and the local combiners c_{kl} must satisfy the following condition (called right stochastic matrix):

$$\sum_{l \in \mathcal{N}_{k,i-1}} c_{kl} = 1, \quad \forall k \tag{3.6}$$

3.2 Mean Transient Analysis

The analysis and derivation of the combined interaction in model (3.5) is a very challenging task and requires different manipulations which is referred to as the Global Model. The global representation of the aforementioned variables is given as follows [4]

$$
\begin{aligned}
\boldsymbol{\psi}^i &\triangleq col\{\boldsymbol{\psi}_1^{(i)}, ..., \boldsymbol{\psi}_N^{(i)}\} \quad (NM \times 1) \\
\boldsymbol{\phi}^{i-1} &\triangleq col\{\boldsymbol{\phi}_1^{(i-1)}, ..., \boldsymbol{\phi}_N^{(i-1)}\} \quad (NM \times 1) \\
\boldsymbol{U}_i &\triangleq dia\{\boldsymbol{u}_{1,i}, ..., \boldsymbol{u}_{N,i}\} \quad (N \times NM) \\
\boldsymbol{d}_i &\triangleq col\{\boldsymbol{d}_1(i), ..., \boldsymbol{d}_N(i)\} \quad (N \times 1) \\
\boldsymbol{v}_i &\triangleq col\{\boldsymbol{v}_1(i), ..., \boldsymbol{v}_N(i)\} \quad (N \times 1) \\
D &\triangleq diag\{\mu_1 I_M, \mu_2 I_M, ..., \mu_N I_M\} \quad (NM \times NM)
\end{aligned} \tag{3.7}
$$

where $\boldsymbol{\psi}^i$ is the global weight update (adaptive or aggregative step) or estimate, $\boldsymbol{\phi}^{i-1}$ is the weight combiner (combine step), $\boldsymbol{U}_i$ is the block regression matrix, $\boldsymbol{d}_i$ is the desired vector (linear measurements), and $\boldsymbol{v}_i$ is the background noise vector at time instant i. The linear model or measurement is declared as

$$\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i} w^o + \boldsymbol{v}_k(i) \tag{3.8}$$

where the background noise $\boldsymbol{v}_k(i)$ is AWGN with mean zero and variance $\sigma_{v,k}^2$. Its global representation is as follows

$$\boldsymbol{d}_i = \boldsymbol{U}_i w^{(o)} + \boldsymbol{v}_i \quad (N \times 1) \tag{3.9}$$

where the optimum weight filter is defined globally by:

$$w^{(o)} = Qw^{o} \quad (NM \times 1) \tag{3.10}$$

The state-space representation of (3.5) in global representation is

$$\begin{aligned} \boldsymbol{\phi}^{i-1} &= G\boldsymbol{\psi}^{i-1} \quad (NM \times 1) \\ \boldsymbol{\psi}^{i} &= \boldsymbol{\phi}^{i-1} + D\boldsymbol{U}_i^{*}(\,\boldsymbol{d}_i - \boldsymbol{U}_i G\boldsymbol{\phi}^{i-1})\,) \quad (NM \times 1) \end{aligned} \tag{3.11}$$

where the global transition matrix $G = C \otimes I_M$ has a size of $NM \times NM$. The global weight error representation can be achieved by subtracting $w^{(o)}$ (Note: $w^{(o)} = Gw^{(o)}$, since it was initially declared that the combination matrix is a right stochastic matrix) from both sides of the state-space representation (3.11). After rearranging, it yields:

$$\tilde{\boldsymbol{\psi}}^{i} = G\tilde{\boldsymbol{\psi}}^{i-1} - D\boldsymbol{U}_i^{*}(\boldsymbol{U}_i G\tilde{\boldsymbol{\psi}}^{i-1} + \boldsymbol{v}_i) \quad (NM \times 1) \tag{3.12}$$

The global weight error representation after segregating the noise from the global weight error becomes

$$\tilde{\boldsymbol{\psi}}^{i} = (\,I_{NM} - D\boldsymbol{U}_i^{*}\boldsymbol{U}_i)\,G\tilde{\boldsymbol{\psi}}^{i-1} - D\boldsymbol{U}_i^{*}\boldsymbol{v}_i \quad (NM \times 1) \tag{3.13}$$

The mean of the weight error is obtained by taking the expectation of (3.13). Assuming independence among the data in $\boldsymbol{u}_{k,i}$ temporally and spatially, we get

$$E\tilde{\boldsymbol{\psi}}^{i} = (\,I_{NM} - DR_u)\,GE\tilde{\boldsymbol{\psi}}^{i-1} \quad (NM \times 1) \tag{3.14}$$

where $R_u = diag\{R_{u,1}, ..., R_{u,N}\}$ is the autocorrelation of the input regressors (It is block diagonal) and $R_{u,k} = E\boldsymbol{u}_{k,i}^{*}\boldsymbol{u}_{k,i}$ is the autocorrelation of node k at time instant i.

Stability in mean can be assured when $\boldsymbol{u}_{k,i}$ is temporally and spatially independent, which yields

$$\lambda(\,(\,I_{NM} - DR_u)\,G)\, < 1 \tag{3.15}$$

In other terms, the eigenvalues must be inside the unit disk (which is the stability condition for a discrete signal). We know that the transition matrix is right stochastic matrix and the previous condition can be written as follows

$$\mid \lambda_{max}(\,(\,I_{NM} - DR_u)\,G)\, \mid < \mid \lambda_{max}(\,I_{NM} - DR_u)\, \mid < 1 \tag{3.16}$$

The DLMS in (3.5) asymptotically reduced to a lower value on average if and only if the step-size is selected to guarantee the following condition

$$0 < \mu_k < \frac{2}{\lambda_{max}(\,R_{u,k})} \tag{3.17}$$

3.3 Mean-Square Analysis

We derive the mean-square error or MSD by computing the weighted square-norm of the weight estimate (or a priori vector) ($\tilde{\psi}^{(i-1)} = w^o - \psi_k^{(i-1)}$) in (3.13) and obtaining its average. We have [4],

$$E \parallel \tilde{\boldsymbol{\psi}}^{i} \parallel_{\Sigma}^{2} = E \parallel \tilde{\boldsymbol{\psi}}^{i-1} \parallel_{\boldsymbol{\Sigma}'}^{2} + E\boldsymbol{v}_i^* \boldsymbol{U}_i D \Sigma D \boldsymbol{U}_i^* \boldsymbol{v}_i \tag{3.18}$$

$$\begin{aligned} \boldsymbol{\Sigma}' &= G^* \Sigma G - G^* \Sigma D \boldsymbol{U}_i^* \boldsymbol{U}_i G - G^* \boldsymbol{U}_i^* \boldsymbol{U}_i D \Sigma G \\ &+ G^* \boldsymbol{U}_i^* \boldsymbol{U}_i D \Sigma D \boldsymbol{U}_i^* \boldsymbol{U}_i G \quad (NM \times NM) \end{aligned} \tag{3.19}$$

where the weighting matrix $\boldsymbol{\Sigma}'$ consists of random values and is dependent on the data regression which can be selected arbitrarily. Assuming independence of the diagonal input data matrix and taking expectation of weighting matrix (Deterministic form: $\Sigma' = E\boldsymbol{\Sigma}'$), we have

$$E \parallel \tilde{\boldsymbol{\psi}}^{i} \parallel_{\Sigma}^{2} = E \parallel \tilde{\boldsymbol{\psi}}^{i-1} \parallel_{\Sigma'}^{2} + E\boldsymbol{v}_i^* \boldsymbol{U}_i D \Sigma D \boldsymbol{U}_i^* \boldsymbol{v}_i \tag{3.20}$$

$$\begin{aligned} \Sigma' &= G^* \Sigma G - G^* \Sigma D E(\boldsymbol{U}_i^* \boldsymbol{U}_i)\, G - G^* E(\boldsymbol{U}_i^* \boldsymbol{U}_i)\, D \Sigma G \\ &+ G^* E(\boldsymbol{U}_i^* \boldsymbol{U}_i D \Sigma D \boldsymbol{U}_i^* \boldsymbol{U}_i)\, G \quad (NM \times NM) \end{aligned}$$

In order to reformulate the MSD, we let the regressors be based on circular gaussian generators. The block diagonal autocorrelation matrix R_u is decompressed in block diagonal eigenvalue matrix Λ by using a unitary matrix T as follows

$$R_u = T\Lambda T^* \quad (NM \times NM) \tag{3.21}$$

where $\Lambda = diag\{\Lambda_1, ..., \Lambda_N\}$ and Λ_k are diagonal matrices only when the regressors are uncorrelated. In addition, all values are transformed as follows

$$\begin{aligned}
\bar{\boldsymbol{\psi}}^i &= T^* \tilde{\boldsymbol{\psi}}^i \quad (NM \times 1) \\
\bar{\boldsymbol{U}}_i &= \boldsymbol{U}_i T \quad (N \times NM) \\
\bar{G} &= T^* G T \quad (NM \times NM) \\
\bar{\Sigma} &= T^T \Sigma T \quad (NM \times NM) \\
\bar{\Sigma}' &= T^T \Sigma' T \quad (NM \times NM) \\
\bar{D} &= T^T D T = D \quad (NM \times NM)
\end{aligned} \tag{3.22}$$

Eventually, the MSD in (3.20) is also transformed and is given by

$$E \parallel \bar{\boldsymbol{\psi}}^i \parallel^2_{\bar{\Sigma}} = E \parallel \bar{\boldsymbol{\psi}}^{i-1} \parallel^2_{\bar{\Sigma}'} + E\boldsymbol{v}_i^* \bar{\boldsymbol{U}}_i D \bar{\Sigma} D \bar{\boldsymbol{U}}_i^* \boldsymbol{v}_i \tag{3.23}$$

$$\begin{aligned}
\bar{\Sigma}' &= \bar{G}^* \bar{\Sigma} \bar{G} - \bar{G}^* \bar{\Sigma} D E(\bar{\boldsymbol{U}}_i^* \bar{\boldsymbol{U}}_i)\, \bar{G} - \bar{G}^T E(\bar{\boldsymbol{U}}_i^* \bar{\boldsymbol{U}}_i)\, D \bar{\Sigma} \bar{G} \\
&\quad + \bar{G}^* E(\bar{\boldsymbol{U}}_i^* \bar{\boldsymbol{U}}_i D \bar{\Sigma} D \bar{\boldsymbol{U}}_i^* \bar{\boldsymbol{U}}_i)\, \bar{G} \quad (NM \times NM)
\end{aligned} \tag{3.24}$$

The recursive form of the MSD for Gaussian regressors is defined as follows

$$E \parallel \bar{\boldsymbol{\psi}}^i \parallel^2_{\bar{\sigma}} = E \parallel \bar{\boldsymbol{\psi}}^{i-1} \parallel^2_{\bar{F}\bar{\sigma}} + b^T \bar{\sigma} \tag{3.25}$$

$$\bar{F} = (\,\bar{G}^T \odot \bar{G}^{*T})\,[\,I_{N^2M^2} - (\,I_{NM} \odot \Lambda D)$$

$$- (\,\Lambda D \odot I_{NM}) + (\,D \odot D)\mathcal{A}] \quad (N^2M^2 \times N^2M^2)$$

where

$$b = \text{bvec}\{R_v D^2 \Lambda\} \quad (N^2M^2 \times 1)$$

$$R_v = \Lambda \odot I_M \quad (NM \times NM)$$

$$\mathcal{A} = diag\{\mathcal{A}_1, \mathcal{A}_2, ..., \mathcal{A}_N\} \quad (N^2M^2 \times N^2M^2)$$

$$\mathcal{A}_k = col\{\Lambda_1 \otimes \Lambda_l, ..., \lambda_l \lambda_l^T + \gamma\Lambda_l \otimes \Lambda_l, ..., \Lambda_N \otimes \Lambda_l\} \quad (NM^2 \times NM^2)$$

$$\bar{\sigma} = \text{bvec}\{\bar{\Sigma}\} \quad (N^2M^2 \times 1)$$

Assuming the initial condition of the globally optimum filter is $\bar{w}^{(o)} = T^* w^{(o)}$, then the recursive MSD in (3.25) can be rewritten for the time instant i as follows

$$E \parallel \bar{\boldsymbol{\psi}}^i \parallel^2_{\bar{\sigma}} = E \parallel \bar{w}^{(o)} \parallel^2_{\bar{F}^{i+1}\bar{\sigma}} + b^T \left(\sum_{k=0}^{i} \bar{F}^k \right) \bar{\sigma} \qquad (3.26)$$

and for the time instant $i-1$ as

$$E \parallel \bar{\boldsymbol{\psi}}^{i-1} \parallel^2_{\bar{\sigma}} = E \parallel \bar{w}^{(o)} \parallel^2_{\bar{F}^{i}\bar{\sigma}} + b^T \left(\sum_{k=0}^{i-1} \bar{F}^k \right) \bar{\sigma} \qquad (3.27)$$

Now if we subtract (3.27) from (3.26), we have an important recursive form:

$$E \parallel \bar{\boldsymbol{\psi}}^i \parallel^2_{\bar{\sigma}} = E \parallel \bar{\boldsymbol{\psi}}^{i-1} \parallel^2_{\bar{\sigma}} + b^T \bar{F}^i \bar{\sigma} - \parallel \bar{w}^{(o)} \parallel^2_{\bar{F}^i \left(I - \bar{F}\right)\bar{\sigma}} \qquad (3.28)$$

The overall network MSD performance can be obtained by selecting $\bar{\sigma} = \frac{1}{N}\text{bvec}\{I_{NM}\} \triangleq q_\eta$ in (3.28) and averaging the MSD $\eta(i) = \frac{1}{N}E \parallel \bar{\boldsymbol{\psi}}^i \parallel_{\bar{\sigma}}^2$, which results in

$$\eta(i) = \eta(i-1) + b^T \bar{F}^i q_\eta - \parallel \bar{w}^{(o)} \parallel^2_{\bar{F}^i\left(I-\bar{F}\right)q_\eta} \tag{3.29}$$

with initial condition $\eta(-1) = \parallel \bar{w}^{(o)} \parallel^2$. Conversely, the local node performance can be obtained by selecting $\bar{\sigma}_k = \text{bvec}\{\mathbf{0}_{(k-1)M}, I_M, \mathbf{0}_{(N-k)M}\} \triangleq q_{k\eta}$ in (3.28) and partitioning the MSD $\eta_k(i) = E \parallel \bar{\boldsymbol{\psi}}^i \parallel^2_{\bar{\sigma}_k}$, which yields

$$\eta_k(i) = \eta_k(i-1) + b^T \bar{F}^i q_{k\eta} - \parallel \bar{w}^{(o)} \parallel^2_{\bar{F}^i\left(I-\bar{F}\right)q_{k\eta}} \tag{3.30}$$

with initial condition $\eta_k(-1) = \parallel \bar{w}^{(o)} \parallel^2_{\bar{\sigma}_k}$. The aforementioned discussion with regards to the derived performance can be presented in a different way for easy implementation and simulation. Let's re-write (3.28) in a different way as follows [14]

$$\begin{aligned} E \parallel \bar{\boldsymbol{\psi}}^{i+1} \parallel^2_{\bar{\sigma}} &= E \parallel \bar{\boldsymbol{\psi}}^i \parallel^2_{\bar{\sigma}} + \parallel \bar{w}^{(o)} \parallel^2_{\bar{F}'^i\left(\bar{F} - I_{N^2M^2}\right)\bar{\sigma}} \\ &\quad + [\,\bar{F}''^i(\,\bar{F} - I_{N^2M^2})\, + b^T I_{N^2M^2}]\,\bar{\sigma} \end{aligned} \tag{3.31}$$

where

$$\bar{F}'^i = \prod_{m=0}^{i-1} \bar{F}(m) \quad (N^2M^2 \times N^2M^2)$$

$$\bar{F}''^i = \sum_{m=0}^{i-2} b^T(m) \left(\prod_{n=m+1}^{i-1} \bar{F}(n) \right) + b^T I_{N^2M^2} \quad (N^2M^2 \times N^2M^2)$$

The overall network MSD performance can be achieved by selecting $\bar{\sigma} = \frac{1}{N}\text{bvec}\{I_{NM}\} \triangleq q_\eta$ in (3.31) and averaging the MSD $\eta(i) = \frac{1}{N}E \parallel \bar{\boldsymbol{\psi}}^i \parallel^2_{\bar{\sigma}}$, which yields

$$\eta(i) = \eta(i-1) + \parallel \bar{w}^{(o)} \parallel^2_{\bar{F}'^i\left(\bar{F}-I_{N^2M^2}\right)q_\eta}$$

$$+ \left[\, \bar{F}''^i \left(\, \bar{F} - I_{N^2M^2}\right) + b^T I_{N^2M^2}\right] q_\eta \tag{3.32}$$

where

$$\bar{F}'^{i+1} = \bar{F}'^i \bar{F}^i \quad (N^2M^2 \times N^2M^2)$$

$$\bar{F}''^{i+1} = \bar{F}''^i \bar{F}^i + b^T I_{N^2M^2} \quad (N^2M^2 \times N^2M^2)$$

with initial condition $\eta(-1) = \parallel \bar{w}^{(o)} \parallel^2$. Conversely, the local node performance can be achieved by selecting $\bar{\sigma}_k = \text{bvec}\{\mathbf{0}_{(k-1)M}, I_M, \mathbf{0}_{(N-k)M}\} \triangleq q_{k\eta}$ in (3.31) and partitioning the MSD $\eta_k(i) = E \parallel \bar{\boldsymbol{\psi}}^i \parallel^2_{\bar{\sigma}_k}$ which yields

$$\eta_k(i) = \eta_k(i-1) + \parallel \bar{w}^{(o)} \parallel^2_{\bar{F}'^i\left(\bar{F}-I_{N^2M^2}\right)q_{k\eta}}$$

$$+ \left[\, \bar{F}''^i \left(\, \bar{F} - I_{N^2M^2}\right) + b^T I_{N^2M^2}\right] q_{k\eta} \tag{3.33}$$

where

$$\bar{F}'^{i+1} = \bar{F}'^i \bar{F}^i \quad (N^2M^2 \times N^2M^2)$$

$$\bar{F}''^{i+1} = \bar{F}''^i \bar{F}^i + b^T I_{N^2M^2} \quad (N^2M^2 \times N^2M^2)$$

The detailed derivations of the Adapt-Then-Combine (ATC) single-task DLMS are discussed in Algorithm A.1(A) (See Appendix-A).

The theoretical derivations that consider the aforementioned equations of mean transient and mean-square analyses for single-task networks adaptive filters (DLMS) are also shown in details in Algorithm A.1(B) (See Appendix-A).

3.4 Extension of Single-Task Variable Step-Size DLMS to Transform Domain

In [14], a modification to the step-size in traditional single-task networks with square-error algorithm was proposed to obtain a variable step-size single-task LMS which improved the global performance. The VSS-DLMS algorithm is based on the squared error that is inserted in the step-size as explained in [3] for single-node networks and will be extended here to single-task networks. The weight update formula is given by

$$\begin{aligned} \psi_{k,i+1} &= \phi_{k,i} + \mu_k(i) u_{k,i}^* (d_k(i) - u_{k,i}\phi_{k,i}) \\ \mu_k(i+1) &= \alpha\mu_k(i) + \gamma e_k^2(i) \\ \phi_{k,i+1} &= \sum_{l \in \mathcal{N}_k} c_{kl}\psi_{l,i+1} \end{aligned} \tag{3.34}$$

with constants $0 < \alpha < 1$ and $\gamma > 0$. The Global *weight error* vector and *a priori error* for node k are respectively defined by

$$\begin{aligned} \tilde{\boldsymbol{\psi}}^i &= w^o - \boldsymbol{\psi}^{i-1} \quad (M \times 1) \\ \tilde{\boldsymbol{\psi}}^i &= \boldsymbol{U}_i \tilde{\boldsymbol{\psi}}^{i-1} \quad (M \times 1) \end{aligned} \tag{3.35}$$

while the global MSD and EMSE are defined by

$$\begin{aligned} \text{MSD} &= E \parallel \tilde{\boldsymbol{\psi}}^{i} \parallel^{2} \\ \text{EMSE} &= E \parallel \tilde{\boldsymbol{\psi}}^{i} \parallel^{2}_{R_u} = E \mid \boldsymbol{e}^{i} \mid^{2} - \mathcal{V} = \mathcal{E}^{i} - \mathcal{V} \end{aligned} \tag{3.36}$$

where $\mathcal{E}^i$ is the Global Mean Square Error (MSE) vector of size $N \times 1$ which stack the MSE of every node k and $\mathcal{V}$ is the Global background noise vector of size $N \times 1$ of each node is σ_k^2. The mean transient and mean-square analyses for square error algorithm are similar to our previous discussion in Sections (3.2) and (3.3) for DLMS with a little modification to account for the random step-size. Since it contains random data (i.e., input regressors $u_{k,i}$) in its square output term $e_k^2(i)$. The mean transient of weight error in (3.14) is modified and it is given by

$$E\bar{\boldsymbol{\psi}}^{i} = (\, I_{NM} - E\,[\boldsymbol{D}_{i-1}]\,\Lambda)\,GE\bar{\boldsymbol{\psi}}^{i-1} \quad (NM \times 1) \tag{3.37}$$

while the mean-square of weight error in (3.25) is modified as follows [14]

$$\begin{aligned} E \parallel \bar{\boldsymbol{\psi}}^{i} \parallel^{2}_{\bar{\sigma}} &= E \parallel \bar{\boldsymbol{\psi}}^{i-1} \parallel^{2}_{\bar{F}\bar{\sigma}} + b^{T}\bar{\sigma} \\ \bar{F} &= (\, \bar{G}^{*} \odot \bar{G}^{*T})\,[\, I_{N^2M^2} - (\, I_{NM} \odot \Lambda E\,[\boldsymbol{D}_{i-1}]) \\ &\quad - (\, \Lambda E\,[\boldsymbol{D}_{i-1}] \odot I_{NM}) + E\,[\boldsymbol{D}_{i-1} \odot \boldsymbol{D}_{i-1}]\,\mathcal{A}] \quad (N^2M^2 \times N^2M^2) \\ &\textit{where} \\ b &= \text{bcev}\{R_v E\left[\boldsymbol{D}^{2}_{i-1}\right]\Lambda\} \quad (N^2M^2 \times 1) \end{aligned} \tag{3.38}$$

The first and second order moments of step-size in (3.34) are defined by

$$
\begin{aligned}
E\left[\boldsymbol{D}_i\right] &= \alpha E\left[\boldsymbol{D}_{i-1}\right] + \gamma \mathcal{E}^{i-1} \quad (N^2M^2 \times N^2M^2) \\
E\left[\boldsymbol{D}_i^2\right] &= \alpha^2 E\left[\boldsymbol{D}_{i-1}^2\right] + 2\alpha\gamma E\left[\boldsymbol{D}_{i-1}\right]\mathcal{E}^{i-1} + \gamma^2 \mathcal{E}^{2,i-1} \quad (N^2M^2 \times N^2M^2)
\end{aligned} \tag{3.39}
$$

The details of the variable step-size DLMS (VSSLMS) algorithm for single-task networks are given in Algorithm A.2(A) (See Appendix-A).

The theoretical derivations of the VSSLMS consider the aforementioned equations for mean transient and mean-square analyses for the single-task networks are outlined (See Algorithm A.2(B) in Appendix-A).

In what follows, we propose and develop several VSSDLMS algorithms to enhance and boost the performance of DLMS. The list of our VSSDLMS algorithms variants are:

1. Transform Domain DLMS (TDLMS)

2. Discrete Cosine Transform DLMS (DCTLMS)

3. Transform Domain Variable Step-Size DLMS (TDVSS)

4. New Variable Step-Size Transform-Domain DLMS (NVSTDLMS)

5. Optimal Variable Step-Size DLMS (VSSTDLMS)

3.4.1 Transform-Domain DLMS (TDLMS)

Our first algorithm converts the input regressor from time domain to transform domain to decorrelate or reduce the correlation of the input. It is shown to provide better MSD performance as discussed by previous authors [5–10] in case of single-node and can be crafted for single-task networks. The weight-update formula is presented by

$$
\begin{aligned}
\psi_{k,i+1} &= \phi_{k,i} + \frac{\mu_k}{\delta + \sigma_{k,i}^2} u_{k,i}^* (\, d_k(i) - u_{k,i}\phi_{k,i}) \\
\sigma_{k,i+1}^2 &= \beta \sigma_{k,i}^2 + (\, 1 - \beta)\, |u_{k,i}|^2 \\
\phi_{k,i+1} &= \sum_{l \in \mathcal{N}_k} c_{kl} \psi_{l,i+1}
\end{aligned}
\tag{3.40}
$$

where $u_{k,i}$ is the transform domain of the input signal, such as DCT, DFT, or DWT for node k and iteration i.

Next, we derive the first and second order moments of step-size by combining the step-size and power estimate in (3.40) as follows:

$$
\mu_k^{'}(i) = \frac{\mu_k}{\delta + \sigma_{k,i}^2} \tag{3.41}
$$

and the global step-size became $D_i \triangleq diag\{\mu_1^{'}(i) I_M, \mu_2^{'}(i) I_M, ..., \mu_N^{'}(i) I_M\}$.

The first and second order moments, $E\left[\boldsymbol{D}_i\right]$ and $E\left[\boldsymbol{D}_i^2\right]$, of the global step-size are defined for each node k by

$$
\begin{aligned}
E\left[\boldsymbol{\mu}_k^{'}(i)\right] &= E\left[\mu_k\right] E\left[\frac{1}{\delta+\boldsymbol{\sigma}_{k,i}^2}\right] \\
E\left[\boldsymbol{\mu}_k^{'2}(i)\right] &= E\left[\mu_k^2\right] E\left[\left(\frac{1}{\delta+\boldsymbol{\sigma}_{k,i}^2}\right)^2\right]
\end{aligned}
\tag{3.42}
$$

which can be re-written as (by using Jensen's Inequality as described in appendix-C)

$$
\begin{aligned}
E\left[\boldsymbol{\mu}_k^{'}(i)\right] &= \begin{cases} \leq \frac{E[\mu_k]}{E\left[\delta+\boldsymbol{\sigma}_{k,i}^2\right]}, & \text{for convex function} \\ \geq \frac{E[\mu_k]}{E\left[\delta+\boldsymbol{\sigma}_{k,i}^2\right]}, & \text{for concave function} \end{cases} \\
&= \frac{\mu_k}{\delta+E\left[\boldsymbol{\sigma}_{k,i}^2\right]}
\end{aligned}
\tag{3.43}
$$

$$
\begin{aligned}
E\left[\boldsymbol{\mu}_k^{'2}(i)\right] &= \begin{cases} \leq \frac{E\left[\mu_k^2\right]}{E\left[\left(\delta+\boldsymbol{\sigma}_{k,i}^2\right)^2\right]}, & \text{for convex function} \\ \geq \frac{E\left[\mu_k^2\right]}{E\left[\left(\delta+\boldsymbol{\sigma}_{k,i}^2\right)^2\right]}, & \text{for concave function} \end{cases} \\
&= \frac{\mu_k^2}{\delta^2+2\delta E\left[\boldsymbol{\sigma}_{k,i}^2\right]+E\left[\boldsymbol{\sigma}_{k,i}^4\right]}
\end{aligned}
\tag{3.44}
$$

In this work, and especially for the simulations, we use a different approach to solve (3.42) by manipulating the second terms as follows (assume δ^2 to be very small),

$$E\left[\frac{1}{\boldsymbol{\sigma}^2_{k,i}+\delta}\right] = E\left[\frac{\boldsymbol{\sigma}^2_{k,i}-\delta}{\boldsymbol{\sigma}^4_{k,i}-\delta^2}\right]$$
$$= E\left[\frac{1}{\boldsymbol{\sigma}^2_{k,i}} - \frac{\delta}{\boldsymbol{\sigma}^4_{k,i}}\right]$$
$$= \begin{cases} \leq \frac{1}{E\left[\boldsymbol{\sigma}^2_{k,i}\right]} - \frac{\delta}{E\left[\boldsymbol{\sigma}^4_{k,i}\right]}, & \text{for convex function} \\ \geq \frac{1}{E\left[\boldsymbol{\sigma}^2_{k,i}\right]} - \frac{\delta}{E\left[\boldsymbol{\sigma}^4_{k,i}\right]}, & \text{for concave function} \end{cases} \tag{3.45}$$

$$E\left[\left(\frac{1}{\delta+\boldsymbol{\sigma}^2_{k,i}}\right)^2\right] = E\left[\left(\frac{1}{\boldsymbol{\sigma}^2_{k,i}} - \frac{\delta}{\boldsymbol{\sigma}^4_{k,i}}\right)^2\right]$$
$$= E\left[\frac{1}{\boldsymbol{\sigma}^4_{k,i}} + \frac{\delta^2}{\boldsymbol{\sigma}^8_{k,i}} - \frac{2\delta}{\boldsymbol{\sigma}^6_{k,i}}\right]$$
$$= \begin{cases} \leq \frac{1}{E\left[\boldsymbol{\sigma}^4_{k,i}\right]} + \frac{\delta^2}{E\left[\boldsymbol{\sigma}^8_{k,i}\right]} - \frac{2\delta}{E\left[\boldsymbol{\sigma}^6_{k,i}\right]}, & \text{if convex} \\ \geq \frac{1}{E\left[\boldsymbol{\sigma}^4_{k,i}\right]} + \frac{\delta^2}{E\left[\boldsymbol{\sigma}^8_{k,i}\right]} - \frac{2\delta}{E\left[\boldsymbol{\sigma}^6_{k,i}\right]}, & \text{if concave} \end{cases} \tag{3.46}$$

Hence, the solutions of the second approach in (3.42) are

$$E\left[\boldsymbol{\mu}'_k(i)\right] = \mu_k\left[\frac{1}{E\left[\boldsymbol{\sigma}^2_{k,i}\right]} - \frac{\delta}{E\left[\boldsymbol{\sigma}^4_{k,i}\right]}\right]$$
$$E\left[\boldsymbol{\mu}'^2_k(i)\right] = \mu^2_k\left[\frac{1}{E\left[\boldsymbol{\sigma}^4_{k,i}\right]} + \frac{\delta^2}{E\left[\boldsymbol{\sigma}^8_{k,i}\right]} - \frac{2\delta}{E\left[\boldsymbol{\sigma}^6_{k,i}\right]}\right] \tag{3.47}$$

The different moments of the power estimates are derived by considering (3.40):

$$
\begin{aligned}
E\left[\boldsymbol{\sigma}^2_{k,i}\right] &= \beta E\left[\boldsymbol{\sigma}^2_{k,i-1}\right] + (1-\beta)\,\sigma^2_k \\
E\left[\boldsymbol{\sigma}^4_{k,i}\right] &= \beta^2 E\left[\boldsymbol{\sigma}^4_{k,i-1}\right] + 2\beta(1-\beta)\,\sigma^2_k E\left[\boldsymbol{\sigma}^2_{k,i-1}\right] + (1-\beta)^2\,\sigma^4_k \\
E\left[\boldsymbol{\sigma}^6_{k,i}\right] &= \beta^3 E\left[\boldsymbol{\sigma}^6_{k,i-1}\right] + 3\beta^2(1-\beta)\,\sigma^2_k E\left[\boldsymbol{\sigma}^4_{k,i-1}\right] + 3\beta(1-\beta)^2\,\sigma^4_k E\left[\boldsymbol{\sigma}^2_{k,i-1}\right] \\
&\quad + (1-\beta)^3\,\sigma^6_k \\
E\left[\boldsymbol{\sigma}^8_{k,i}\right] &= \beta^4 E\left[\boldsymbol{\sigma}^8_{k,i-1}\right] + 4\beta^3(1-\beta)\,\sigma^2_k E\left[\boldsymbol{\sigma}^6_{k,i-1}\right] + 6\beta^2(1-\beta)^2\,\sigma^4_k E\left[\boldsymbol{\sigma}^4_{k,i-1}\right] \\
&\quad + 4\beta(1-\beta)^3\,\sigma^6_k E\left[\boldsymbol{\sigma}^2_{k,i-1}\right] + (1-\beta)^4\,\sigma^8_k
\end{aligned}
\tag{3.48}
$$

where σ^2_k is the variance of the input regressor for node k.

The variable step-size transform domain DLMS (TDLMS) for single-task networks is summarized in Algorithm A.3 (A) (See Appendix-A).

The theoretical derivations of the variable step-size transform domain DLMS (TDLMS) that consider the aforementioned equations for mean transient and mean-square analyses for single-task networks are outlined in Algorithm A.3(B) (See Appendix-A).

3.4.2 Discrete Cosine Transform DLMS (DCTLMS)

Our second proposed method of VSSDLMS algorithm is based on the power estimator of the current input as well as the previous input signal. The estimate of the input signal is inserted in the global step-size as explained in [12] in the case of single-node networks. This approach is extended to single-task networks in the same manner. The weight update formula is given by

$$\begin{aligned}
\psi_{k,i+1} &= \phi_{k,i} + \mu_{k,i} u_{k,i}^{*} (d_k(i) - u_{k,i}\phi_{k,i}) \\
\mu_{k,i+1} &= \beta\mu_{k,i} + \gamma(1-\beta) \left(\frac{1}{\delta + \frac{1}{L} u_{k,i}^{'*} u_{k,i}^{'}} \right) \\
\phi_{k,i+1} &= \sum_{l \in \mathcal{N}_k} c_{kl}\psi_{l,i+1}
\end{aligned} \tag{3.49}$$

where $\mu_{k,i+1}$ is a vector of size $M \times 1$, $u_{k,i}^{'} = col\{u_{k,i}, u_{k,i-1}, ..., u_{k,i-(L-1)}\}$ is the regressor Matrix of past L coefficients, L is the number of past considered regressors, $0 \leq \beta \leq 1$ and $0 \leq \gamma \leq 1$ are some constants, and δ is a small constant to avoid an infinite solution. The first and second order moments of the variable step-size in (3.49) become

$$\begin{aligned}
E\left[\boldsymbol{\mu}_{k,i+1}\right] &= \beta E\left[\boldsymbol{\mu}_{k,i}\right] + \gamma(1-\beta) E\left[\frac{1}{\delta + \frac{1}{L}\boldsymbol{u}_{k,i}^{'*}\boldsymbol{u}_{k,i}^{'}}\right] \quad (M \times 1) \\
E\left[\boldsymbol{\mu}_{k,i+1}^{2}\right] &= \beta^2 E\left[\boldsymbol{\mu}_{k,i}^{2}\right] + 2\beta\gamma(1-\beta) E\left[\boldsymbol{\mu}_{k,i}\right] E\left[\frac{1}{\delta + \frac{1}{L}\boldsymbol{u}_{k,i}^{'*}\boldsymbol{u}_{k,i}^{'}}\right] \\
&\quad + \gamma^2(1-\beta)^2 E\left[\left(\frac{1}{\delta + \frac{1}{L}\boldsymbol{u}_{k,i}^{'*}\boldsymbol{u}_{k,i}^{'}}\right)^2\right] \quad (M \times 1)
\end{aligned} \tag{3.50}$$

Assuming random Gaussian input regressors in (3.50), we can write [12]

$$\begin{aligned}
E\left[\frac{1}{\delta + \frac{1}{L}\boldsymbol{u}_{k,i}^{'*}\boldsymbol{u}_{k,i}^{'}}\right] &= \frac{L}{\sigma_k^2(L-2)} - \delta\frac{L^2}{\sigma_k^4(L-4)(L-2)} \quad (M \times 1) \\
E\left[\left(\frac{1}{\delta + \frac{1}{L}\boldsymbol{u}_{k,i}^{'*}\boldsymbol{u}_{k,i}^{'}}\right)^2\right] &= \frac{L^2}{\sigma_k^4(L-4)(L-2)} - 2\delta\frac{L^3}{\sigma_k^6(L-6)(L-4)(L-2)} \\
&\quad + \delta^2\frac{L^4}{\sigma_k^8(L-8)(L-6)(L-4)(L-2)} \quad (M \times 1)
\end{aligned} \tag{3.51}$$

The variable step-size algorithm based on DCTLMS for single-task networks is summarized in Algorithm A.4(A) (See Appendix-A).

The theoretical derivations of the variable step-size algorithm based on DCTLMS that considers the aforementioned equations for mean transient and mean-square analyses for the single-task filter are outlined in Algorithm A.4(B) (See Appendix-A). The simulations are discussed in section 3.5.

3.4.3 Transform-Domain Variable Step-Size DLMS (TD-VSS)

In the third algorithm, the constant global step-size is changed with time and continuously learned as described in [9] for single-node networks. This approach can be developed for single-task networks in the same manner. Initially, the algorithm selects the maximum step-size due to the large mismatch in output error. After convergence, it switches to minimum step-size when the output error mismatch becomes very small due to convergence. The weight-update formula is given by

$$\psi_{k,i+1} = \phi_{k,i} + \frac{\mu_k(i)}{\delta + \sigma^2_{k,i}} u^*_{k,i}(\, d_k(i) - u_{k,i}\phi_{k,i})$$

$$\sigma^2_{k,i+1} = \beta\sigma^2_{k,i} + (\,1-\beta)\,|u_{k,i}|^2$$

$$A_k(i) = \alpha\mu_k(i) + \frac{\gamma}{L}\sum_{j=i-(\,L-1)}^{i} e^2_k(j)$$

$$\mu_k(i+1) = \begin{cases} A_k(i), & \text{if } i = kL \text{ and } A_k(i) \in (\,\mu_{k,min}, \mu_{k,max}) \\ \\ \mu_{k,max}, & \text{if } i = kL \text{ and } A_k(i) \geq \mu_{k,max} \\ \\ \mu_{k,min}, & \text{if } i = kL \text{ and } A_k(i) \leq \mu_{k,min} \\ \\ \mu_k(i), & \text{if } i \neq kL \end{cases}$$

$$\phi_{k,i+1} = \sum_{l \in \mathcal{N}_k} c_{kl} \psi_{l,i+1} \tag{3.52}$$

Next, we derive the first and second order moments of the step-size by combining the step-size and power estimate in (3.52) as follows

$$\mu_k^{'}(i) = \frac{\mu_k(i)}{\delta + \sigma_{k,i}^2} \tag{3.53}$$

therefore, the global step-size becomes $D_i \triangleq diagonal\{\mu_1^{'}(i) I_M, \mu_2^{'}(i) I_M, ..., \mu_N^{'}(i) I_M\}$.

The first and second order moments, $E\left[\boldsymbol{D}_i\right]$ and $E\left[\boldsymbol{D}_i^2\right]$, of global step-size are defined for each node k by

$$\begin{aligned} E\left[\boldsymbol{\mu}_k^{'}(i)\right] &= \frac{E\left[\boldsymbol{\mu}_k(i)\right]}{E\left[\delta + \boldsymbol{\sigma}_{k,i}^2\right]} \\ E\left[\boldsymbol{\mu}_k^{'2}(i)\right] &= \frac{E\left[\boldsymbol{\mu}_k^2(i)\right]}{E\left[\left(\delta + \boldsymbol{\sigma}_{k,i}^2\right)^2\right]} \end{aligned} \tag{3.54}$$

which should be formulated as follows

$$
\begin{aligned}
E\left[\boldsymbol{\mu}_k^{'}(i)\right] &= \frac{E\left[\boldsymbol{\mu}_k(i)\right]}{\delta + E\left[\boldsymbol{\sigma}_{k,i}^2\right]} \\
E\left[\boldsymbol{\mu}_k^{'2}(i)\right] &= \frac{E\left[\boldsymbol{\mu}_k^2(i)\right]}{\delta^2 + 2\delta E\left[\boldsymbol{\sigma}_{k,i}^2\right] + E\left[\boldsymbol{\sigma}_{k,i}^4\right]}
\end{aligned}
\tag{3.55}
$$

In this work and especially in simulation, we use a different approach to solve the mean and variance of step-size as described in the aforementioned section when we discussed the TDLMS approach. The solutions are

$$
\begin{aligned}
E\left[\boldsymbol{\mu}_k^{'}(i)\right] &= E\left[\boldsymbol{\mu}_k(i)\right]\left[\frac{1}{E\left[\boldsymbol{\sigma}_{k,i}^2\right]} - \frac{\delta}{E\left[\boldsymbol{\sigma}_{k,i}^4\right]}\right] \\
E\left[\boldsymbol{\mu}_k^{'2}(i)\right] &= E\left[\boldsymbol{\mu}_k^2(i)\right]\left[\frac{1}{E\left[\boldsymbol{\sigma}_{k,i}^4\right]} + \frac{\delta^2}{E\left[\boldsymbol{\sigma}_{k,i}^8\right]} - \frac{2\delta}{E\left[\boldsymbol{\sigma}_{k,i}^6\right]}\right]
\end{aligned}
\tag{3.56}
$$

The first and second order moments of original step-size in (3.55) are derived by considering the condition in (3.52). Then, we have

$$
\begin{aligned}
E\left[\boldsymbol{\mu}_k(i)\right] &= \alpha E\left[\boldsymbol{\mu}_k(i-1)\right] + \frac{\gamma}{L}\sum_{j=i-(L-1)}^{i} E\left[\boldsymbol{e}_k^2(j)\right] \\
E\left[\boldsymbol{\mu}_k^2(i)\right] &= \alpha^2 E\left[\boldsymbol{\mu}_k^2(i-1)\right] \\
&\quad + 2\alpha\frac{\gamma}{L}\sum_{j=i-(L-1)}^{i} E\left[\boldsymbol{e}_k^2(j)\right] E\left[\boldsymbol{\mu}_k(i-1)\right] \\
&\quad + \frac{\gamma^2}{L^2}\sum_{j=i-(L-1)}^{i} E\left[\boldsymbol{e}_k^4(j)\right]
\end{aligned}
\tag{3.57}
$$

The transform domain variable step-size DLMS (TDVSS) for single-task networks is outlined in Algorithm A.5(A) (See Appendix-A).

The theoretical derivations of the transform domain variable step-size DLMS (TD-VSS) algorithm that considers the aforementioned equations for mean transient and mean-square analyses for the single-task networks are outlined in Algorithm A.5(B) (See Appendix-A). The simulations are discussed in section 3.5.

3.4.4 New Variable Step-Size Transform-Domain DLMS (NVSTDLMS)

Our fourth algorithm depends on the ratio of the output errors as analyzed in [13] for single-node networks which will be developed for single-task networks. It does not require the assumption of uncorrelated AWGN. The output error is defined by

$$\varepsilon_k(i) = \alpha \varepsilon_k(i-1) + e_k^2(i-1) \tag{3.58}$$

The weight-update formula is presented by

$$\psi_{k,i+1} = \phi_{k,i} + \frac{\mu_k(i)}{\delta + \sigma_{k,i}^2} u_{k,i}^* (\, d_k(i) - u_{k,i}\phi_{k,i})$$

$$\sigma_{k,i+1}^2 = \beta \sigma_{k,i}^2 + (\, 1 - \beta)\, |u_{k,i}|^2$$

$$\mu_k(i+1) = \begin{cases} \frac{\varepsilon_{1,k}(i)}{\varepsilon_{2,k}(i)}, & \text{if } \mu_k(i+1) \in (\,\mu_{k,min}, \mu_{k,max}) \\ \mu_{k,max}, & \text{if } \mu_k(i+1) \geq \mu_{k,max} \\ \mu_{k,min}, & \text{if } \mu_k(i+1) \leq \mu_{k,min} \end{cases}$$

$$\phi_{k,i+1} = \sum_{l \in \mathcal{N}_k} c_{kl} \psi_{l,i+1} \tag{3.59}$$

where $0 \leq \alpha_1 \leq \alpha_2 \leq 1$ in $\varepsilon_{1,k}(i)$ and $\varepsilon_{2,k}(i)$ are control constants. Next, we derive the mean and variance of the step-size by combining the step-size and the power estimate in (3.59) as follows:

$$\mu_k^{'}(i) = \frac{\mu_k(i)}{\delta + \sigma_{k,i}^2} \tag{3.60}$$

therefore, the global step-size becomes $D_i \triangleq diag\{\mu_1^{'}(i)I_M, \mu_2^{'}(i)I_M, ..., \mu_N^{'}(i)I_M\}$. The first and second order moments, $E\left[\boldsymbol{D}_i\right]$ and $E\left[\boldsymbol{D}_i^2\right]$, of global step-size are defined for each node k by

$$\begin{aligned} E\left[\boldsymbol{\mu}_k^{'}(i)\right] &= \frac{E\left[\boldsymbol{\mu}_k(i)\right]}{E\left[\delta + \boldsymbol{\sigma}_{k,i}^2\right]} \\ E\left[\boldsymbol{\mu}_k^{'2}(i)\right] &= \frac{E\left[\boldsymbol{\mu}_k^2(i)\right]}{E\left[\left(\delta + \boldsymbol{\sigma}_{k,i}^2\right)^2\right]} \end{aligned} \tag{3.61}$$

which should be formulated as follows

$$E\left[\boldsymbol{\mu}_k^{'}(i)\right] = \frac{E\left[\boldsymbol{\mu}_k(i)\right]}{\delta + E\left[\boldsymbol{\sigma}_{k,i}^2\right]}$$

$$E\left[\boldsymbol{\mu}_k'^2(i)\right] = \frac{E\left[\boldsymbol{\mu}_k^2(i)\right]}{\delta^2 + 2\delta E\left[\boldsymbol{\sigma}_{k,i}^2\right] + E\left[\boldsymbol{\sigma}_{k,i}^4\right]} \tag{3.62}$$

In this work and especially in the simulation, we use a different approach to solve the mean and variance of the step-size as described in the aforementioned section when we discussed the TDLMS approach. The solutions are

$$\begin{aligned} E\left[\boldsymbol{\mu}_k'(i)\right] &= E\left[\boldsymbol{\mu}_k(i)\right]\left[\frac{1}{E\left[\boldsymbol{\sigma}_{k,i}^2\right]} - \frac{\delta}{E\left[\boldsymbol{\sigma}_{k,i}^4\right]}\right] \\ E\left[\boldsymbol{\mu}_k'^2(i)\right] &= E\left[\boldsymbol{\mu}_k^2(i)\right]\left[\frac{1}{E\left[\boldsymbol{\sigma}_{k,i}^4\right]} + \frac{\delta^2}{E\left[\boldsymbol{\sigma}_{k,i}^8\right]} - \frac{2\delta}{E\left[\boldsymbol{\sigma}_{k,i}^6\right]}\right] \end{aligned} \tag{3.63}$$

The first and second order moments of the original step-size in (3.62) are derived by considering (3.58) and the declaration in (3.59). It becomes

$$\begin{aligned} E\left[\boldsymbol{\mu}_k(i)\right] &= \frac{E\left[\boldsymbol{\varepsilon}_{1,k}(i)\right]}{E\left[\boldsymbol{\varepsilon}_{2,k}(i)\right]} \\ E\left[\boldsymbol{\mu}_k^2(i)\right] &= \frac{E\left[\boldsymbol{\varepsilon}_{1,k}^2(i)\right]}{E\left[\boldsymbol{\varepsilon}_{2,k}^2(i)\right]} \end{aligned} \tag{3.64}$$

which should be formulated as follows

$$\begin{aligned} E\left[\boldsymbol{\mu}_k(i)\right] &= \frac{\alpha_1 E\left[\boldsymbol{\varepsilon}_{1,k}(i-1)\right] + E\left[\boldsymbol{e}_k^2(i)\right]}{\alpha_2 E\left[\boldsymbol{\varepsilon}_{2,k}(i-1)\right] + E\left[\boldsymbol{e}_k^2(i)\right]} \\ E\left[\boldsymbol{\mu}_k^2(i)\right] &= \frac{\alpha_1^2 E\left[\boldsymbol{\varepsilon}_{1,k}^2(i-1)\right] + 2\alpha_1 E\left[\boldsymbol{\varepsilon}_{1,k}(i-1)\right] E\left[\boldsymbol{e}_k^2(i)\right] + E\left[\boldsymbol{e}_k^4(i)\right]}{\alpha_2^2 E\left[\boldsymbol{\varepsilon}_{2,k}^2(i-1)\right] + 2\alpha_2 E\left[\boldsymbol{\varepsilon}_{2,k}(i-1)\right] E\left[\boldsymbol{e}_k^2(i)\right] + E\left[\boldsymbol{e}_k^4(i)\right]} \end{aligned} \tag{3.65}$$

The variable step-size algorithm based on NVSTDLMS for single-task networks is summarized in Algorithm A.6(A) (See Appendix-A).

The theoretical derivations of the variable step-size algorithm based on NVST-DLMS that consider the aforementioned equations for mean transient and mean-square analyses for single-task networks are outlined in Algorithm A.6(B) (See Appendix-A). The simulations are discussed in section 3.5.

3.4.5 Optimal Variable Step-Size Transform Domain DLMS (VSSTDLMS)

In the fifth algorithm, the AWGN is uncorrelated as per the assumption that was introduced by [10] which can be extended to single-task networks. The weight-update formula is presented by

$$\begin{aligned}
\psi_{k,i+1} &= \phi_{k,i} + \frac{\mu_k(i)}{\delta + \sigma_{k,i}^2} u_{k,i}^* (\, d_k(i) - u_{k,i}\phi_{k,i}) \\
\sigma_{k,i+1}^2 &= \beta\sigma_{k,i}^2 + (\, 1 - \beta)\, |u_{k,i}|^2 \\
\rho_k(i+1) &= \gamma\rho_k(i) + (\, 1 - \gamma)\, \left(e_k^2(i)\hat{u}_{k,i}^T \hat{u}_{k,i} \right) \\
\eta_k(i+1) &= \gamma\eta_k(i) + (\, 1 - \gamma)\, \big(e_k(i) e_k(i-1) \big) \\
\mu_k(i+1) &= \frac{\big|\eta_k(i+1)\big|}{\rho_k(i+1)} \\
\phi_{k,i+1} &= \sum_{l \in \mathcal{N}_k} c_{kl}\psi_{l,i+1}
\end{aligned} \tag{3.66}$$

where $\hat{u}_{k,i} = V_{k,i}^{-1} u_{k,i}$ with $V_{k,i}^2 = \text{diag}\{\sigma_{1,k,i}^2, \sigma_{2,k,i}^2, \ldots, \sigma_{M,k,i}^2\}$.

Next, we derive the mean and variance of step-size by combining the step-size and the power estimate in (3.66) as follows:

$$\mu_k^{'}(i) = \frac{\left|\eta_k(i)\right| / \rho_k(i)}{\delta + \sigma_{k,i}^2} \tag{3.67}$$

therefore, the global step-size becomes $D_i \triangleq diag\{\mu_1^{'}(i)I_M, \mu_2^{'}(i)I_M, ..., \mu_N^{'}(i)I_M\}$. The mean $E\left[\boldsymbol{D}_i\right]$ and the variance $E\left[\boldsymbol{D}_i^2\right]$ of global step-size are defined for node k by

$$\begin{aligned} E\left[\boldsymbol{\mu}_k^{'}(i)\right] &= \frac{E\left[\left|\boldsymbol{\eta}_k(i)\right| / \boldsymbol{\rho}_k(i)\right]}{E\left[\delta + \boldsymbol{\sigma}_{k,i}^2\right]} \\ E\left[\boldsymbol{\mu}_k^{'2}(i)\right] &= \frac{E\left[\left(\left|\boldsymbol{\eta}_k(i)\right| / \boldsymbol{\rho}_k(i)\right)^2\right]}{E\left[\left(\delta + \boldsymbol{\sigma}_{k,i}^2\right)^2\right]} \end{aligned} \tag{3.68}$$

which can be simplified as

$$\begin{aligned} E\left[\boldsymbol{\mu}_k^{'}(i)\right] &= \frac{E\left[\left|\boldsymbol{\eta}_k(i)\right|\right] / E\left[\boldsymbol{\rho}_k(i)\right]}{\delta + E\left[\boldsymbol{\sigma}_{k,i}^2\right]} \\ E\left[\boldsymbol{\mu}_k^{'2}(i)\right] &= \frac{E\left[\left|\boldsymbol{\eta}_k(i)\right|^2\right] / E\left[\boldsymbol{\rho}_k^2(i)\right]}{\delta^2 + 2\delta E\left[\boldsymbol{\sigma}_{k,i}^2\right] + E\left[\boldsymbol{\sigma}_{k,i}^4\right]} \end{aligned} \tag{3.69}$$

In this work and especially in the simulation, we use a different approach to solve the first and second order moments of the step-size as described in the aforementioned section when we discussed the TDLMS approach. The solutions are

$$E\left[\boldsymbol{\mu}_k^{'}(i)\right] = E\left[\left|\boldsymbol{\eta}_k(i)\right|\right] / E\left[\boldsymbol{\rho}_k(i)\right]\left[\frac{1}{E\left[\boldsymbol{\sigma}_{k,i}^2\right]} - \frac{\delta}{E\left[\boldsymbol{\sigma}_{k,i}^4\right]}\right]$$

$$E\left[\boldsymbol{\mu}_k^{'2}(i)\right] = E\left[\left|\boldsymbol{\eta}_k(i)\right|^2\right] / E\left[\boldsymbol{\rho}_k^2(i)\right]\left[\frac{1}{E\left[\boldsymbol{\sigma}_{k,i}^4\right]} + \frac{\delta^2}{E\left[\boldsymbol{\sigma}_{k,i}^8\right]} - \frac{2\delta}{E\left[\boldsymbol{\sigma}_{k,i}^6\right]}\right] \tag{3.70}$$

The first and second order moments of the original step-size in (3.69) are derived by considering the condition in (3.66). It became,

$$\begin{aligned}
E\left[\boldsymbol{\eta}_k(i)\right] &= \gamma E\left[\boldsymbol{\eta}_k(i-1)\right] + (1-\gamma)\left(E\left[\boldsymbol{e}_k(i)\right]E\left[\boldsymbol{e}_k(i-1)\right]\right) \\
E\left[\boldsymbol{\eta}_k^2(i)\right] &= \gamma^2 E\left[\boldsymbol{\eta}_k^2(i-1)\right] \\
&\quad + 2\gamma(1-\gamma)E\left[\boldsymbol{\eta}_k(i-1)\right]\left(E\left[\boldsymbol{e}_k(i)\right]E\left[\boldsymbol{e}_k(i-1)\right]\right) \\
&\quad + (1-\gamma)^2\left(E\left[\boldsymbol{e}_k^2(i)\right]E\left[\boldsymbol{e}_k^2(i-1)\right]\right)
\end{aligned} \tag{3.71}$$

and

$$\begin{aligned}
E\left[\boldsymbol{\rho}_k(i)\right] &= \gamma E\left[\boldsymbol{\rho}_k(i-1)\right] + (1-\gamma)\left(E\left[\boldsymbol{e}_k^2(i)\right]\sigma_k^2 / E\left[\boldsymbol{\sigma}_{k,i}^2\right]\right) \\
E\left[\boldsymbol{\rho}_k^2(i)\right] &= \gamma^2 E\left[\boldsymbol{\rho}_k^2(i-1)\right] \\
&\quad + 2\gamma(1-\gamma)E\left[\boldsymbol{\rho}_k(i-1)\right]\left(E\left[\boldsymbol{e}_k^2(i)\right]\sigma_k^2 / E\left[\boldsymbol{\sigma}_{k,i}^2\right]\right) \\
&\quad + (1-\gamma)^2\left(E\left[\boldsymbol{e}_k^4(i)\right]\sigma_k^4 / E\left[\boldsymbol{\sigma}_{k,i}^4\right]\right)
\end{aligned} \tag{3.72}$$

The optimal variable step-size transform domain DLMS (VSSTDLMS) for single-task networks is summarized in Algorithm A.7(A) (See Appendix-A).

The theoretical derivations of the optimal variable step-size transform domain (VSSTDLMS) algorithm that considers the aforementioned equations for mean transient and mean-square analyses for single-task networks are outlined in Algorithm A.7(B) (See Appendix-A).

3.5 Single-Task Networks Simulations

In this section, we discuss several practical experiments to validate the theoretical derivations in the aforementioned sections. In the all simulations, the input regressors are represented as:

$$u_{k,i} = \text{col}\{u_k(i), u_k(i-1), \ldots, u_k(i-(M-1))\} \tag{3.73}$$

The input regressors are considered to have zero-mean and a variance of one ($\sigma^2_{u,k} = 1$).

In order to produce quality plots, 1000 independent experiments were conducted and averaged. The transient plots were produced by running the single-task networks learning process for less than 6000 iterations. The desired data $d_k(i)$ was produced according to the model (3.8) with the unknown single-task optimum filter $w^o = \text{col}\{1, 1, \ldots, 1\}$ of length $(M \times 1)$ with $M = 2$ taps. Figure 3.2 represents the topology of the single-task network. The AWGN noise level of the multi-task network is shown in Figure 4.3

The experimental parameters (i.e., standard LMS, traditional TDLMS [5], DCTLMS [8], TDVSS [9], VSSTDLMS [10], and NVSTDLMS [11], VSSLMS [3]) are provided and are adjusted based on the parameters settings in [9, 11] as listed in Table 3.1. In our experiments, we used the DCT transform domain except for *traditional LMS* and *VSSLMS* algorithms which are in the time domain. The combiner matrix (right stochastic matrix) was selected according to the Metropolis rule [4] is as follows:

$$C = \begin{bmatrix} 1/4 & 1/4 & 0 & 1/4 & 0 & 1/4 & 0 \\ 1/4 & 1/2 & 1/4 & 0 & 0 & 0 & 0 \\ 0 & 1/4 & 1/4 & 0 & 1/4 & 1/4 & 0 \\ 1/4 & 0 & 0 & 1/2 & 0 & 0 & 1/4 \\ 0 & 0 & 1/4 & 0 & 1/2 & 0 & 1/4 \\ 1/4 & 0 & 1/4 & 0 & 0 & 1/4 & 1/4 \\ 0 & 0 & 0 & 1/4 & 1/4 & 1/4 & 1/4 \end{bmatrix} \tag{3.74}$$

The performance of the single-task networks algorithms for two cases, namely: single node and collaborative nodes when using combiner matrix are shown in Figures 3.4, 3.5, 3.6, 3.7, 3.8, and 3.9. In each case, the experiment considered theoretical and practical simulations and the results matched well. The setup of the experiments was selected carefully to carry out a fair comparison among the algorithms.

The best performance was for VSSLMS (See Figure 3.6) and DCTLMS (See Figure 3.9) algorithms with MSDs of -62 dB and -59 dB, respectively, when the nodes

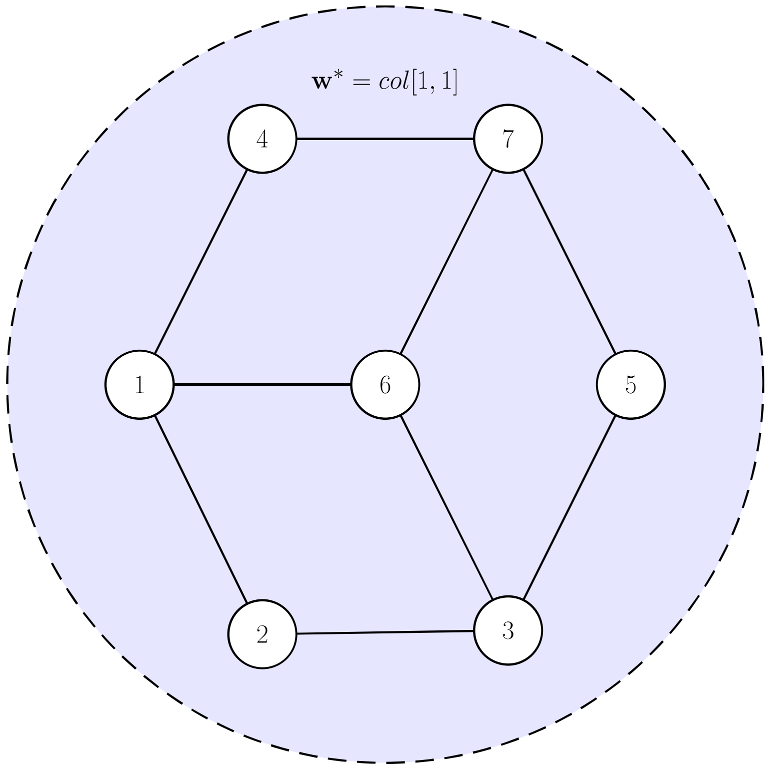

Figure 3.2: Single-Task Network Topology with $N = 7$ Nodes. The optimum filter is $w^o = \text{col}\{1, 1\}$.

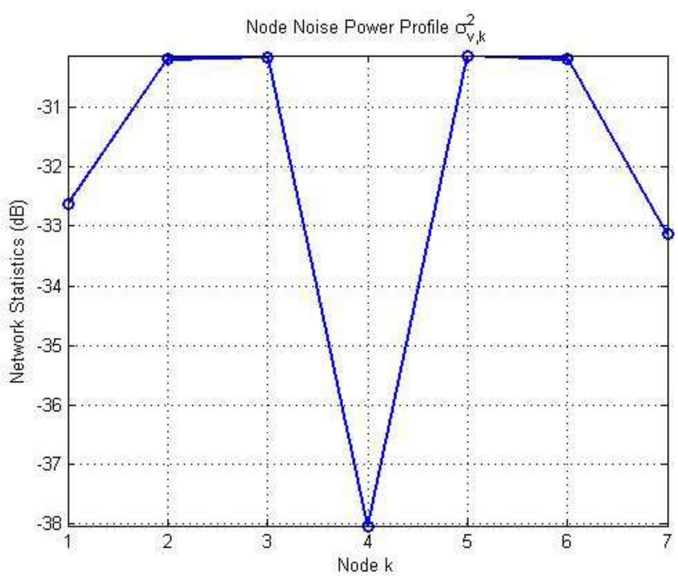

Figure 3.3: Noise Power Profile.

Table 3.1: Parameter Settings For Single-Task Networks Algorithms

DLMS	**NVSTDLMS**
$\mu = 5 \times 10^{-2}$	$\mu = 5 \times 10^{-2}$ $\beta = 0.9$ $\delta = 2.5 \times 10^{-2}$

TDLMS
$\mu_{max} = 1 \times 10^{-1}$ $\alpha_1 = 1 - 7 \times 10^{-3}$ $\mu_{min} = 5 \times 10^{-2}$ $\alpha_2 = 1 - 10^{-5}$ $\beta = 0.9$ $\delta = 5 \times 10^{-2}$

VSSLMS	**DCTLMS**
$\mu = 5 \times 10^{-2}$ $\alpha = 0.9985$ $\gamma = 8 \times 10^{-3}$	$\gamma = 8 \times 10^{-3}$ $\beta = 0.9985$ $\delta = 8 \times 10^{-4}$ $L = 10$

TDVSS
$\mu_{max} = 1 \times 10^{-1}$ $\alpha = 0.9985$ $\beta = 0.9$ $L = 10$ $\mu_{min} = 5 \times 10^{-2}$ $\gamma = 8 \times^{-3}$ $\delta = 2.5 \times 10^{-2}$

collaborate. In the case of a single node, the MSD degraded to -55 dB and -52 dB, respectively. The gain in the MSD in the case of collaboration over a single node was -7 dB.

The performances of the DLMS, TDLMS, TDVSS, and NVSTDLMS algorithms were very similar with a MSD of -50.5 dB for the case of collaborative nodes and a degraded MSD of -43.5 dB in the case of a single node mode. In these experiments, the gain in MSD of collaboration was constant at -7 dB.

3.6 Discussion of The Results

For single-task networks, numerous algorithms of variable step-size transform domain for diffusion strategies have been formulated to improve the MSD performance of the traditional DLMS single-task networks. Mainly two algorithms have shown superior performance, namely: the DCTLMS and the VSSLMS algorithms, which converged around -60 dB and -63 dB, respectively. Where as, the other al-

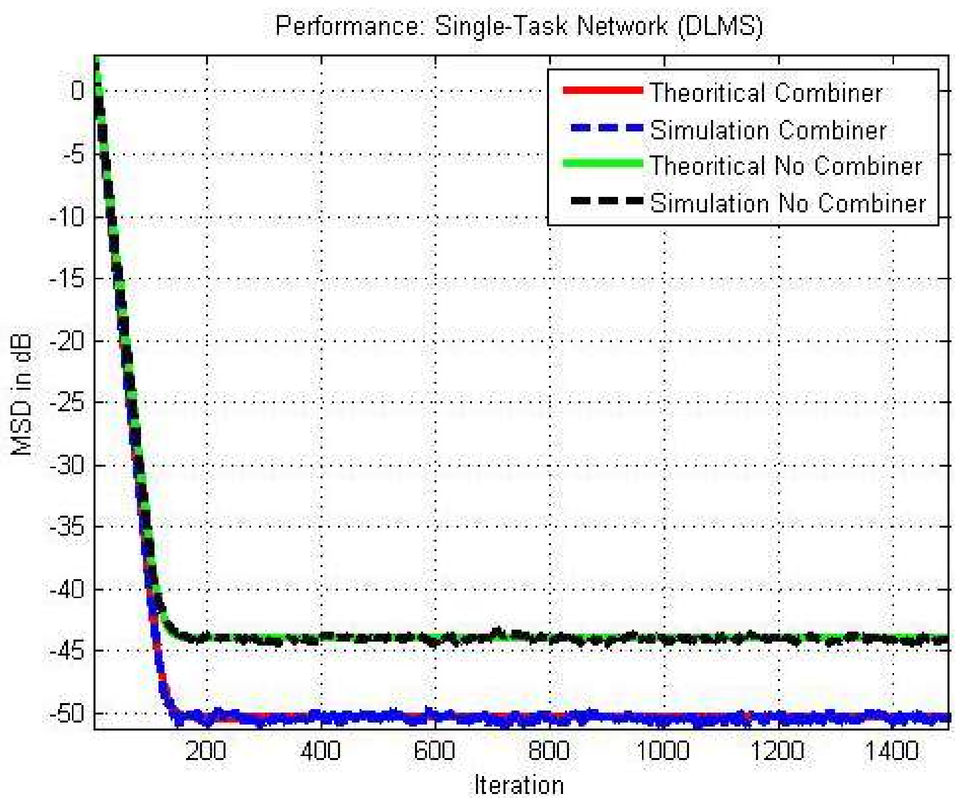

Figure 3.4: Performance of Single-Task Networks using DLMS.

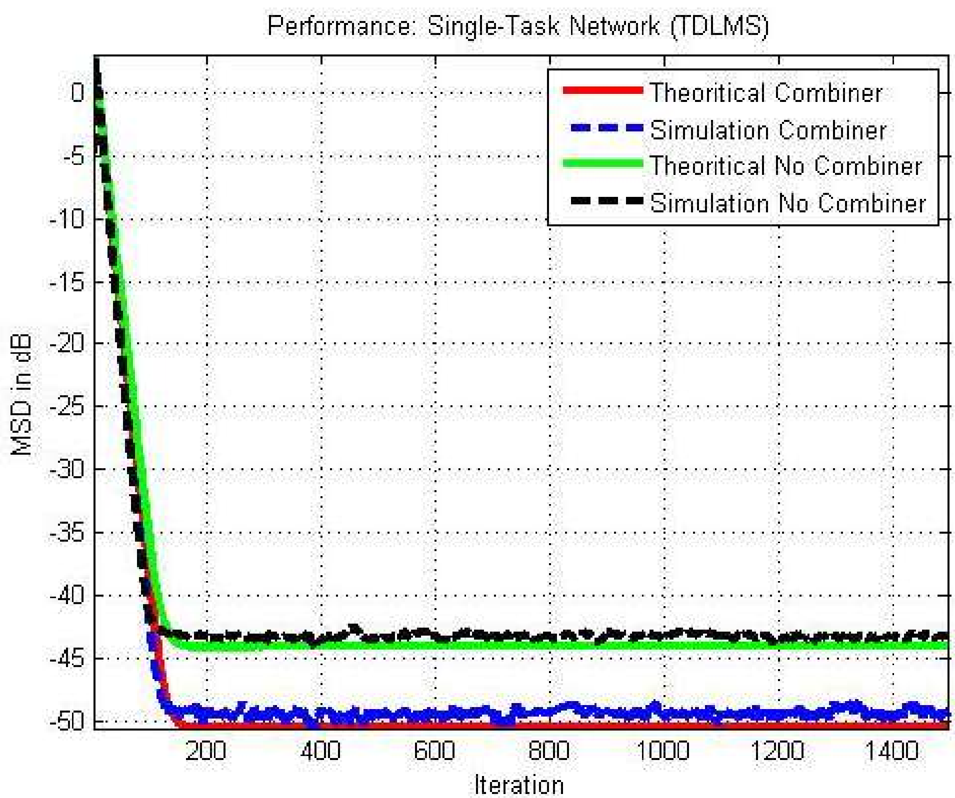

Figure 3.5: Performance of Single-Task Networks using TDLMS.

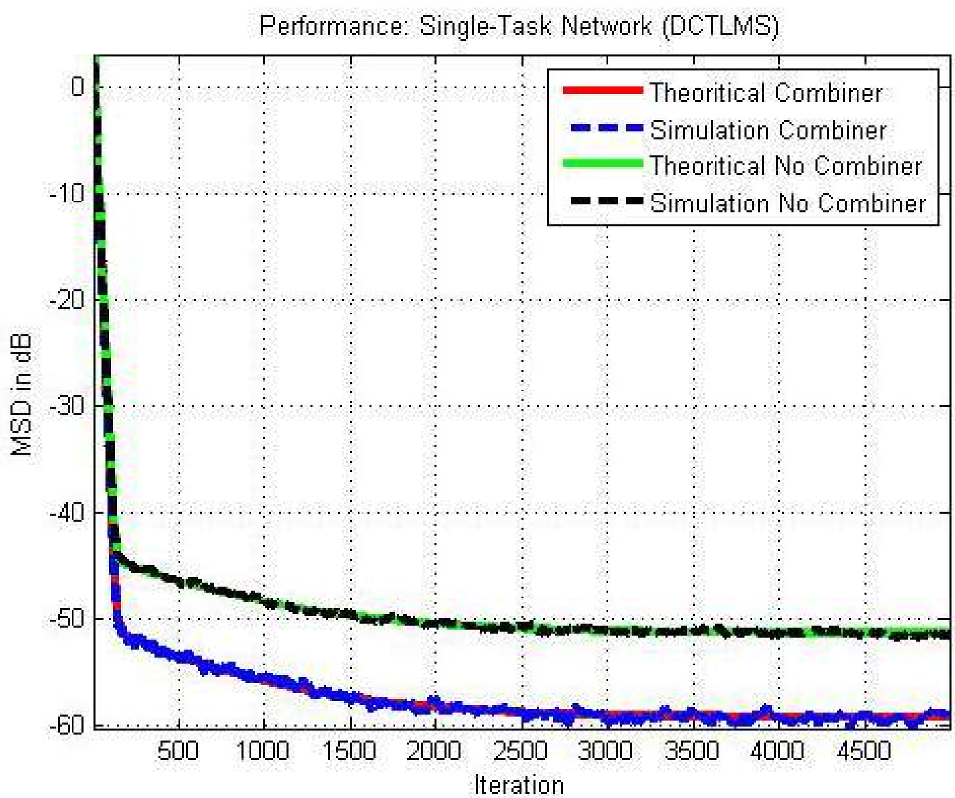

Figure 3.6: Performance of Single-Task Networks using DCTDLMS.

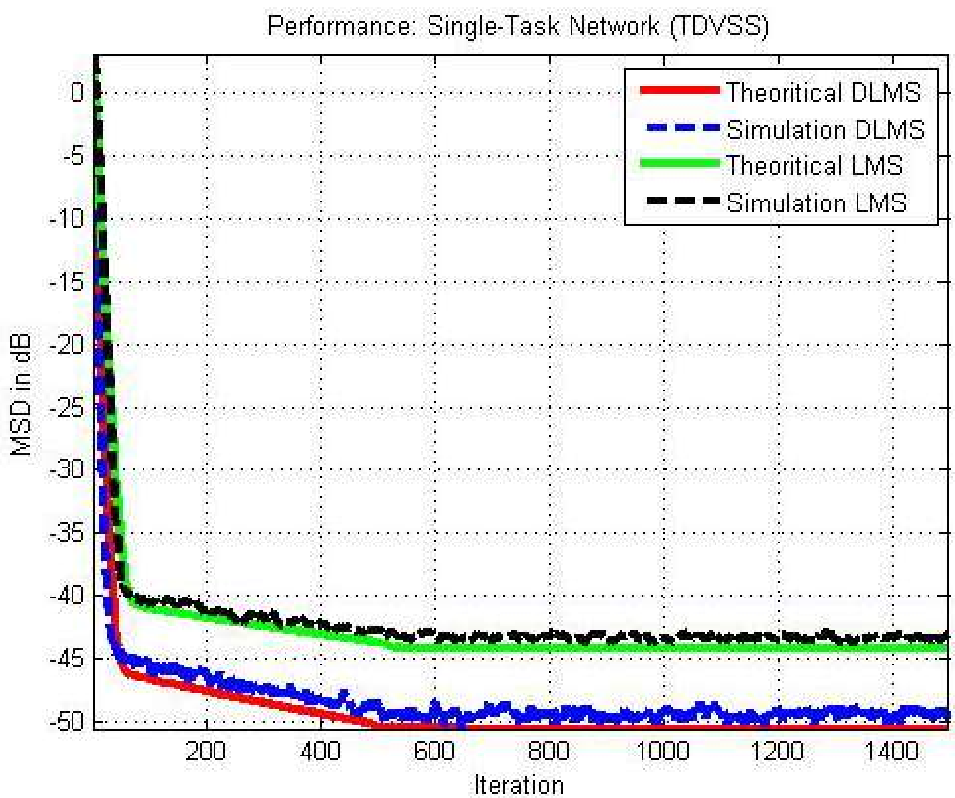

Figure 3.7: Performance of Single-Task Networks using TDVSS.

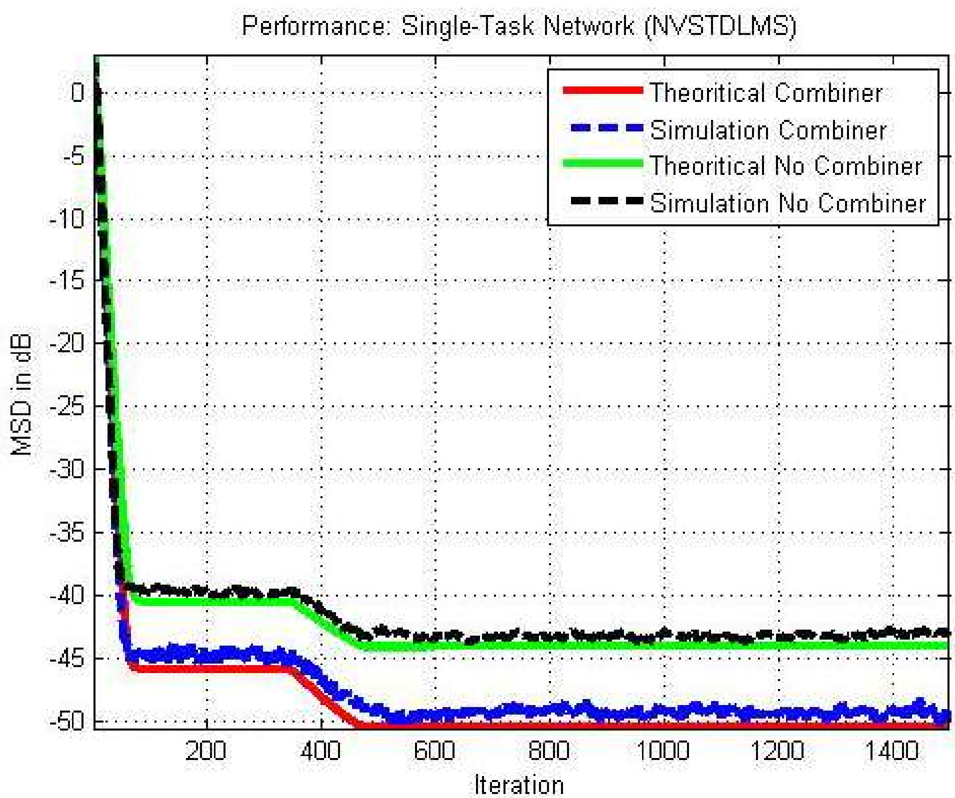

Figure 3.8: Performance of Single-Task Networks using NVSTDLMS.

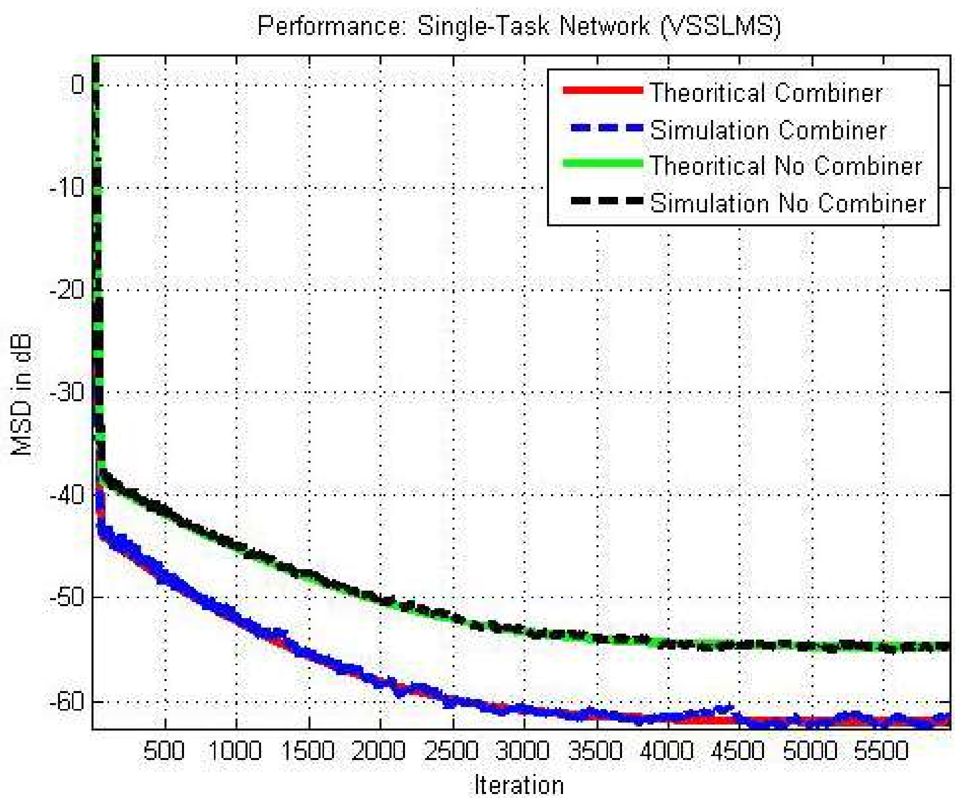

Figure 3.9: Performance of Single-Task Networks using VSSDLMS.

gorithms converged at -50 dB as shown in Figure 3.10. When the convergence factor α got changed from 8×10^{-3} to 1×10^{-3}, the MSD performance of DCTLMS and VSSLMS algorithms was improved further to -70 dB and -72 dB, respectively, as illustrated in Figure 3.11 and Figure 3.12. Typically, we can also reduce the power estimate factor β to improve the performance curve even more.

Although, the performance of DLMS has been shown to be very close to the TDLMS, TDVSS, and NVSTDLMS algorithms nevertheless, we can readjust the parameter settings to achieve better performance in the case of transform domain algorithms. It is worth mentioning that in this chapter, the parameter settings have been adjusted to similar work by previous authors for fairness of comparison.

3.7 Summary

In this chapter, we have developed many variable step-size transform domain algorithms for single-task networks. The theoretical equations have been derived as well. Our simulations have shown that VSSDLMS and DCTLMS outperform the conventional DLMS.

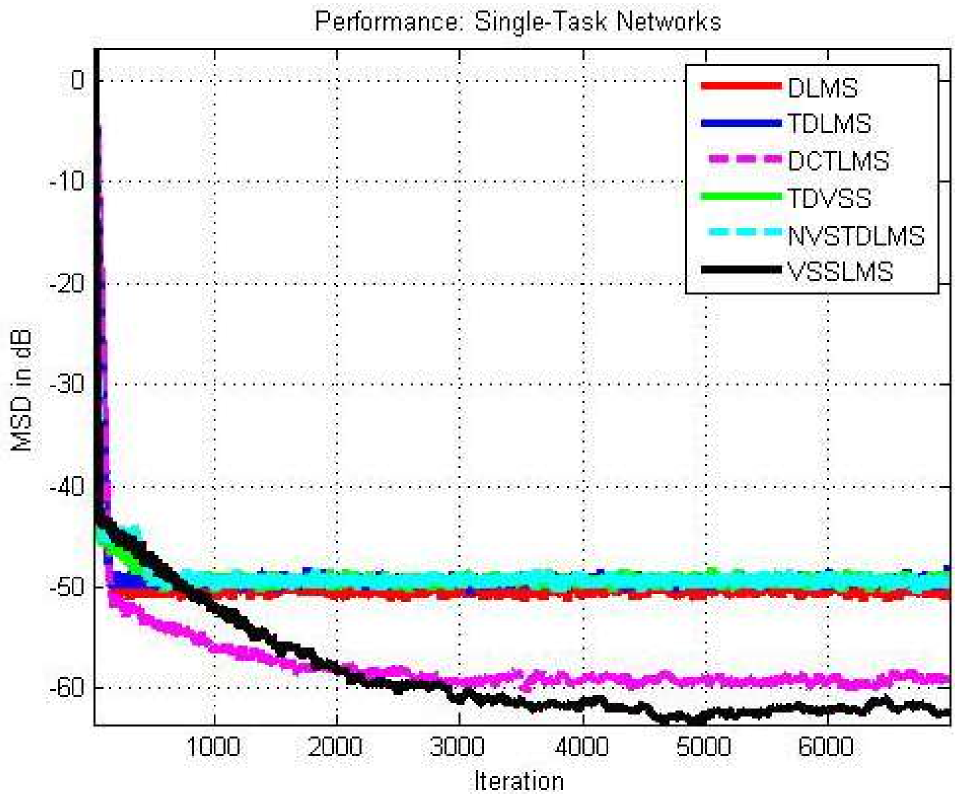

Figure 3.10: Performance Comparison of All Single-Task Networks Algorithms.

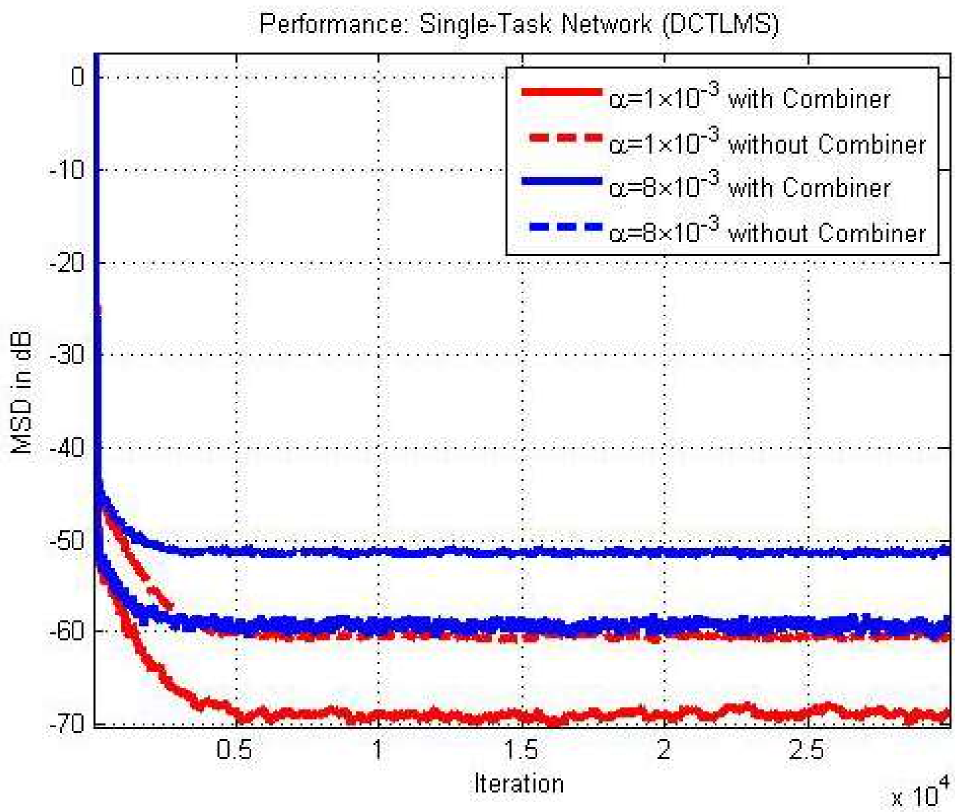

Figure 3.11: Performance Comparison of Single-Task Networks using DCTLMS when α Get Replaced By Lower Value where Other Settings Preserved Fixed.

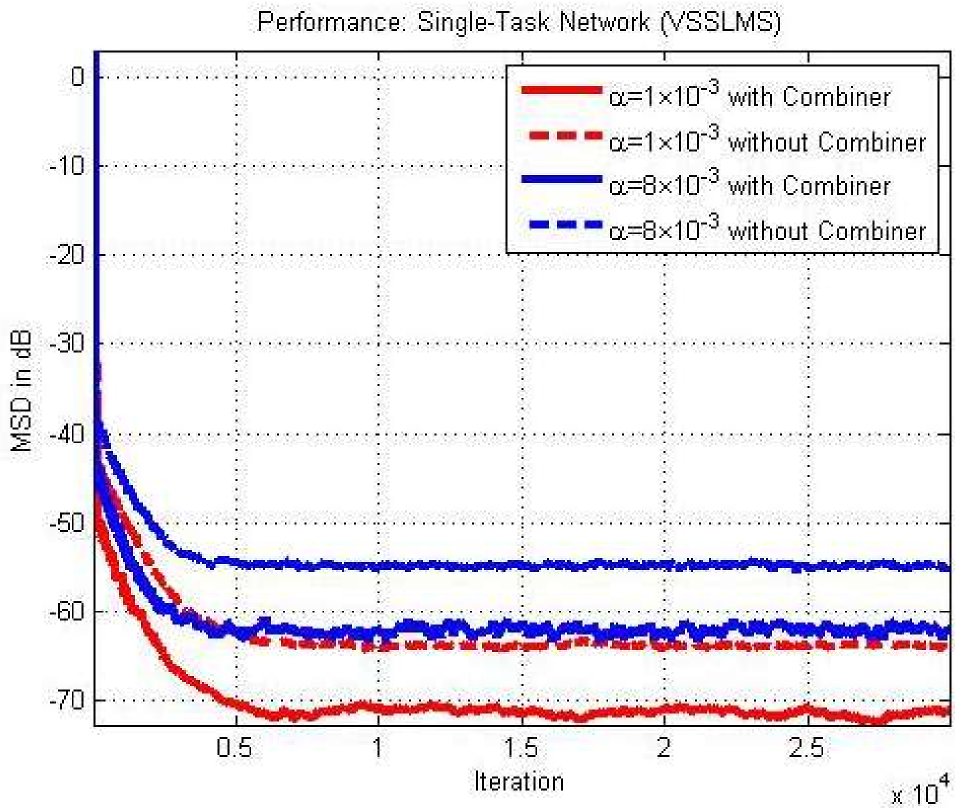

Figure 3.12: Performance Comparison of Single-Task Networks using VSSLMS DLMS when α Get Replaced By Lower Value where Other Settings Preserved Fixed.

CHAPTER 4

TRANSFORM DOMAIN VARIABLE STEP-SIZE DLMS ALGORITHMS (MULTI-TASK NETWORKS)

The challenge in analyzing multi-task networks can be addressed by converting single-task networks with the help of adaptive combiners or adaptive clusters. As a matter of fact, the global mean-square deviation is formulated by considering the individual changes in the mean with minor manipulations as discussed in previous chapters. Variable step-size methods are considered in the sequel and consequently, the global performance is discussed in more details.

The estimation problem of multi-task networks can be obtained by reformulating the single-task problem in multi-task environment as shown in Figure 4.1. Here,

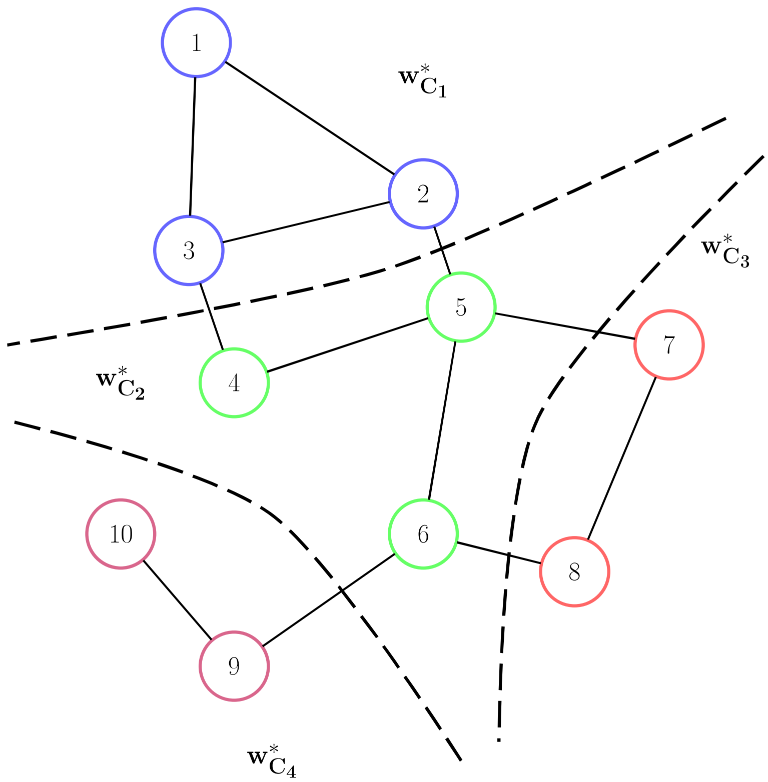

Figure 4.1: A Multi-Task Network with 4 Clusters and 10 Nodes.

local cost function depends on the cluster architecture and is selected to be the mean-square error. This function is defined for node k as follows [2]

$$J_k(w_{C(k)}) = E\{|d_k(i) - u_{k,i}w^o_{C(k)}|^2\} \tag{4.1}$$

where $w_{C(k)}$ is the optimum filter victor for cluster $C(k)$ to which node k belongs.

4.1 Adaptive Combiners

Multi-task adaptive networks can easily be analyzed when an adaptive combiner is employed in the environment of single-task networks as described in the previous chapter. In [17], the authors proposed that an adaptive combination rule which can be inserted in the weight update equation (See Algorithm A.1 (A) in Appendix-A) and is highlighted here as follows:

$$
\begin{aligned}
q_{k,i} &= u^{*}_{k,i}(\, d_k(i) - u_{k,i}\psi_{k,i+1}) \\
c_{kl}(i+1) &= \frac{\parallel \psi_{k,i+1} + \mu_{k,i}q_{k,i} - \psi_{l,i+1} \parallel^{-2}}{\sum_{j\in\mathcal{N}_k} \parallel \psi_{k,i+1} + \mu_{k,i}q_{k,i} - \psi_{j,i+1} \parallel^{-2}}
\end{aligned} \tag{4.2}
$$

for all nodes l in the neighbour of node k, $\mathcal{N}_k$. We will reformulate the aforementioned equation in a different way as:

$$
c_{kl}(i+1) = \frac{\parallel \psi'_{k,i+1} - \psi_{l,i+1} \parallel^{-2}}{\sum_{j\in\mathcal{N}_k} \parallel \psi'_{k,i+1} - \psi_{j,i+1} \parallel^{-2}} \tag{4.3}
$$

where $\psi'_{k,i+1} = \psi_{k,i+1} + \mu_k q_{k,i}$. Now, we add and subtract the optimum filter w^o and modify the exponent in the equation of the adaptive combiner. This results in

$$
c_{kl}(i+1) = \frac{\left(\parallel \left(\psi'_{k,i+1} - w^o\right) - \left(\psi_{l,i+1} - w^o\right) \parallel^2\right)^{-1}}{\sum_{j\in\mathcal{N}_k} \left(\parallel \left(\psi'_{k,i+1} - w^o\right) - \left(\psi_{j,i+1} - w^o\right) \parallel^2\right)^{-1}} \tag{4.4}
$$

Next, we expand the squared expressions in the numerator and the denominator which yield the theoretical adaptive clustering in terms of the mean and the mean-square of the weight error. The resulting expected value $\boldsymbol{E}\boldsymbol{C}_{i+1}$ at $i+1$ iteration is then obtained as:

$$E\boldsymbol{c}_{kl}(i+1) = \frac{\left(E \parallel \left(\psi'_{k,i+1} - w^o\right) - \left(\psi_{l,i+1} - w^o\right) \parallel^2\right)^{-1}}{\sum_{j\in\mathcal{N}_k}\left(E \parallel \left(\psi'_{k,i+1} - w^o\right) - \left(\psi_{j,i+1} - w^o\right) \parallel^2\right)^{-1}} \tag{4.5}$$

$$E\boldsymbol{c}_{kl}(i+1) = \frac{\left(\eta'_k(i+1) + \eta_l(i+1) - 2\zeta'_k(i+1)\zeta_l(i+1)\right)^{-1}}{\sum_{j\in\mathcal{N}_k}\left(\eta'_k(i+1) + \eta_j(i+1) - 2\zeta'_k(i+1)\zeta_j(i+1)\right)^{-1}} \tag{4.6}$$

where $\eta_l(i+1)$ and $\eta'_k(i+1)$ are the mean-squares of the local node obtained from $E \parallel \bar{\boldsymbol{\psi}}^{i+1} \parallel^2$ and $E \parallel \bar{\boldsymbol{\psi}}'^{i+1} \parallel^2$, respectively, while $\zeta_l(i+1)$ and $\zeta'_k(i+1)$ are the mean of the weight error of the local node obtained from $E\bar{\boldsymbol{\psi}}^{i+1}$ and $E\bar{\boldsymbol{\psi}}'^{i+1}$, respectively. The mean and mean square derivations of the weight error for the adaptive combiner $E\boldsymbol{C}_{i+1}$ are discussed in the following sections.

4.2 Mean Transient Analysis

In this section, we derive two equations of the mean transient that appeared in (4.6). We consider the analysis for single-task DLMS in (3.13) in which the adaptive combiner is random data. The result is

$$E\bar{\boldsymbol{\psi}}^i = \left(I_{NM} - D\Lambda\right) E\bar{\boldsymbol{G}}_i E\bar{\boldsymbol{\psi}}^{i-1} \tag{4.7}$$

or in a compact form

$$E\bar{\boldsymbol{\psi}}^i = \bar{F}_1 E\bar{\boldsymbol{\psi}}^{i-1} \tag{4.8}$$

$$\bar{F}_1 = (I_{NM} - D\Lambda)\, E\bar{\boldsymbol{G}}_i$$

which can be retrieved recursively in the following form

$$\begin{aligned}
E\bar{\boldsymbol{\psi}}^i &= \bar{F}_1 E\bar{\boldsymbol{\psi}}^{i-1} \\
\bar{F}_1 E\bar{\boldsymbol{\psi}}^{i-1} &= \bar{F}_1^2 E\bar{\boldsymbol{\psi}}^{i-2} \\
\bar{F}_1^2 E\bar{\boldsymbol{\psi}}^{i-2} &= \bar{F}_1^3 E\bar{\boldsymbol{\psi}}^{i-3} \\
&\vdots \\
\bar{F}_1^{i-1} E\bar{\boldsymbol{\psi}}^{1} &= \bar{F}_1^{i} E\bar{\boldsymbol{\psi}}^{0} \\
\bar{F}_1^{i} E\bar{\boldsymbol{\psi}}^{0} &= \bar{F}_1^{i+1} E\bar{\boldsymbol{w}}^{(o)}
\end{aligned} \tag{4.9}$$

$$\tag{4.10}$$

When the local adaptive multi-task filter is initialized with the optimum filter, it leads to

$$E\bar{\boldsymbol{\psi}}^i = \bar{F}_1^{i+1} \bar{w}^{(o)} \tag{4.11}$$

$$E\bar{\boldsymbol{\psi}}^{i-1} = \bar{F}_1^{i} \bar{w}^{(o)}$$

Now, we subtract $E\bar{\boldsymbol{\psi}}^{i-1}$ in (4.11) from $E\bar{\boldsymbol{\psi}}^{i}$, we have

$$E\bar{\boldsymbol{\psi}}^{i} = E\bar{\boldsymbol{\psi}}^{i-1} - \left(I_{NM} - \bar{F}_1\right)\bar{F}_1^i \bar{w}^{(o)} \tag{4.12}$$

Conversely, the local node mean can be obtained by enforcing $\bar{\sigma}_{1,k} = \text{bvec}\{\mathbf{0}_{(k-1)M}, \mathbf{1}_M, \mathbf{0}_{(N-k)M}\} \triangleq q_{1,k\eta}$ in (4.12) and partitioning the mean weight error $\zeta_k(i) = q_{1,k\eta}^T E\bar{\boldsymbol{\psi}}^{i}$ which is useful later, the result is

$$\zeta_k(i) = \zeta_k(i-1) - q_{1,k\eta}^T \left(I_{NM} - \bar{F}_1\right)\bar{F}_1^i \bar{w}^{(o)} \tag{4.13}$$

Similarly, the second equation of the mean weight error $E\bar{\boldsymbol{\psi}}'^{i+1}$ in (4.6) is defined by

$$\zeta_k'(i) = \zeta_k(i) - q_{1,k\eta}^T \left(I_{NM} - \bar{F}_1\right)\bar{F}_1^i \bar{w}^{(o)} \tag{4.14}$$

4.3 Mean-Square Analysis

In this section, we formulate two equations of mean-square transient that appeared before in (4.4). Indeed, we considered the analysis of the local mean-square for single-task DLMS in (3.25) and (3.30) in which the adaptive combiner was random data. The two resultant equations of mean-square $\eta_k(i)$ and $\eta_k^{'}(i)$ are

$$\eta_k(i) = \eta_k(i-1) + b^T \bar{F}^i q_{k\eta} - \parallel \bar{w}^{(o)} \parallel^2_{\bar{F}^i\left(I-\bar{F}\right)q_{k\eta}} \tag{4.15}$$

$$\eta_k^{'}(i) = \eta_k(i) + b^T \bar{F}^i q_{k\eta} - \parallel \bar{w}^{(o)} \parallel^2_{\bar{F}^i\left(I-\bar{F}\right)q_{k\eta}}$$

$$\bar{F} = E\left[\bar{\boldsymbol{G}}^*_{i-1} \odot \bar{\boldsymbol{G}}^{*T}_{i-1}\right]\left[\, I_{N^2M^2} - (\, I_{NM} \odot \Lambda D) - (\, \Lambda D \odot I_{NM}) + (\, D \odot D)\mathcal{A}\right] \quad (N^2M^2 \times N^2M^2)$$

The Multi-Task DLMS (DLMS) with the Adaptive Combiners technique is derived in Algorithm B.1 (A) (See Appendix-B).

The theoretical derivations of the Adaptive Combiner that considers the aforementioned equations for mean transient and mean-square analyses in order to achieve multi-task adaptive filter are discussed in Algorithm B.1 (B) (See Appendix-B).

4.4 Extension of Single-Task Networks to Multi-Task Networks

In what follows, we extend the work in chapter 3 to multi-task networks with adaptive clustering. The list VSSLMS algorithms that we consider here are:

1. Variable Step-Size DLMS (VSSLMS) with Adaptive Combiners
2. Transform Domain DLMS (TDLMS) with Adaptive Combiners
3. Discrete Cosine Transform DLMS (DCTLMS) with Adaptive Combiners

4. Transform Domain Variable Step-Size DLMS (TDVSS) with Adaptive Combiners

5. New Variable Step-Size Transform-Domain DLMS (NVSTDLMS) with Adaptive Combiners

6. Optimal Step-Size Transform Domain DLMS (VSSTDLMS) with Adaptive Combiners

In order to extend these algorithms to achieve multi-tasking in wireless sensor networks (WSN), we must consider an adaptive clustering technique using the ATC topology. The adaptive combiners are shown lead to good performance that can be improved further by variable step size transform-domain algorithms.

4.4.1 Variable Step-Size DLMS (VSSLMS) with Adaptive Combiners

The proposed VSSLMS algorithm is based on the one in [3] using a global step-size and discussed previously in Chapter 3. The weight-update formulas for the single-task networks were

$$\begin{aligned} \psi_{k,i+1} &= \phi_{k,i} + \mu_k(i) u^*_{k,i} (\, d_k(i) - u_{k,i}\phi_{k,i}) \\ \mu_k(i+1) &= \alpha\mu_k(i) + \gamma e_k^2(i) \\ \phi_{k,i+1} &= \sum_{l \in \mathcal{N}_k} c_{kl}\psi_{l,i+1} \end{aligned} \tag{4.16}$$

The adaptive clustering explained in Section 4.1 with the performance analysis in Section 4.3. In multi-task networks, the weight-update formulas become

$$
\begin{aligned}
\psi_{k,i+1} &= \phi_{k,i} + \mu_k(i) u_{k,i}^*(d_k(i) - u_{k,i}\phi_{k,i}) \\
\mu_k(i+1) &= \alpha\mu_k(i) + \gamma e_k^2(i) \\
q_{k,i} &= u_{k,i}^*(d_k(i) - u_{k,i}\psi_{k,i+1}) \\
c_{kl}(i+1) &= \frac{\| \psi_{k,i+1} + \mu_k(i+1)q_{k,i} - \psi_{l,i+1} \|^{-2}}{\sum_{j\in\mathcal{N}_k} \| \psi_{k,i+1} + \mu_k(i+1)q_{k,i} - \psi_{j,i+1} \|^{-2}} \\
\phi_{k,i+1} &= \sum_{l\in\mathcal{N}_k} c_{kl}(i+1)\psi_{l,i+1} \qquad (4.17)
\end{aligned}
$$

The mean transient of weight error in (3.37) is modified since the adaptive combiner and the step-size becomes random. Therefore, we have

$$
E\bar{\boldsymbol{\psi}}^i = (I_{NM} - E[\boldsymbol{D}_{i-1}]\Lambda) E\left[\bar{\boldsymbol{G}}_{i-1}\right] E\bar{\boldsymbol{\psi}}^{i-1} \quad (NM \times 1) \qquad (4.18)
$$

while the mean-square performance of the weight error in (3.38) is modified for the same reasons as follows [14]

$$
\begin{aligned}
E \| \bar{\boldsymbol{\psi}}^i \|_{\bar{\sigma}}^2 &= E \| \bar{\boldsymbol{\psi}}^{i-1} \|_{\bar{F}\bar{\sigma}}^2 + b^T\bar{\sigma} \qquad (4.19) \\
\bar{F} &= E\left[\bar{\boldsymbol{G}}_{i-1}^* \odot E\bar{\boldsymbol{G}}_{i-1}^{*T}\right] [I_{N^2M^2} - (I_{NM} \odot \Lambda E[\boldsymbol{D}_{i-1}]) \\
&\quad - (\Lambda E[\boldsymbol{D}_{i-1}] \odot I_{NM}) + E[\boldsymbol{D}_{i-1} \odot \boldsymbol{D}_{i-1}]\mathcal{A}] \quad (N^2M^2 \times N^2M^2)
\end{aligned}
$$

where

$$
b = \text{bcev}\{R_v E\left[\boldsymbol{D}_{i-1}^2\right]\Lambda\} \quad (N^2M^2 \times 1)
$$

The first and second order moments of variable step-size have been derived in (3.39).

The algorithm of the variable step-size DLMS (VSSLMS) for multi-task networks with adaptive combiners technique is summarized in Algorithm B.2 (A) (See Appendix-B).

The theoretical derivations of the variable step-size DLMS (VSSLMS) with adaptive clustering that considers the aforementioned equations for mean transient and mean-square analyses for the multi-task adaptive filter are discussed in Algorithm B.2 (B) (See Appendix-B).

4.4.2 Transform Domain DLMS (TDLMS) with Adaptive Combiners

The second algorithm was derived in Chapter 3 for single-task networks. The weight-update formulas for single-task network were

$$\begin{aligned}
\psi_{k,i+1} &= \phi_{k,i} + \frac{\mu_k}{\delta + \sigma^2_{k,i}} u^*_{k,i}(\, d_k(i) - u_{k,i}\phi_{k,i}) \\
\sigma^2_{k,i+1} &= \beta\sigma^2_{k,i} + (\, 1-\beta)\, |u_{k,i}|^2 \\
\phi_{k,i+1} &= \sum_{l\in\mathcal{N}_k} c_{kl}\psi_{l,i+1}
\end{aligned} \tag{4.20}$$

In the sequel, the adaptive clustering in section 4.1 is considered to obtain our proposed multi-task networks. The weight-update formulas for multi-task networks become

$$
\begin{aligned}
\psi_{k,i+1} &= \phi_{k,i} + \frac{\mu_k}{\delta + \sigma^2_{k,i}} u^*_{k,i}(\, d_k(i) - u_{k,i}\phi_{k,i}) \\
\sigma^2_{k,i+1} &= \beta \sigma^2_{k,i} + (\, 1 - \beta)\, |u_{k,i}|^2 \\
q_{k,i} &= u^*_{k,i}(\, d_k(i) - u_{k,i}\psi_{k,i+1}) \\
c_{kl}(i+1) &= \frac{\parallel \psi_{k,i+1} + \mu_k q_{k,i} - \psi_{l,i+1} \parallel^{-2}}{\sum_{j \in \mathcal{N}_k} \parallel \psi_{k,i+1} + \mu_k q_{k,i} - \psi_{j,i+1} \parallel^{-2}} \\
\phi_{k,i+1} &= \sum_{l \in \mathcal{N}_k} c_{kl}(i+1)\psi_{l,i+1}
\end{aligned}
\tag{4.21}
$$

The transform domain DLMS (TDLMS) for multi-task networks is summarized in Algorithm B.3 (A) (See Appendix-B).

The theoretical derivations of the variable step-size transform domain DLMS (TDLMS) with adaptive combiners that consider the aforementioned equations for mean transient and mean-square analyses for multi-task networks are outlined in Algorithm B.3 (B) (See Appendix-B).

4.4.3 Discrete Cosine Transform DLMS (DCTLMS) with Adaptive Combiners

The previously proposed variable step-size algorithm based on DCTLMS in (3.49) without adaptive clustering is reconsidered for multi-task networks. In single-task networks, the weight-update formulas were

$$
\psi_{k,i+1} = \phi_{k,i} + \mu_{k,i} u^*_{k,i}(\, d_k(i) - u_{k,i}\phi_{k,i})
$$

$$\mu_{k,i+1} = \beta\mu_{k,i} + \gamma(1-\beta)\left(\frac{1}{\delta + \frac{1}{L}u'^{*}_{k,i}u'_{k,i}}\right)$$

$$\phi_{k,i+1} = \sum_{l\in\mathcal{N}_k} c_{kl}\psi_{l,i+1} \tag{4.22}$$

When adaptive combiners are considered, the weight-update formulas for multi-task networks become

$$\begin{aligned}
\psi_{k,i+1} &= \phi_{k,i} + \mu_{k,i}u^{*}_{k,i}(d_k(i) - u_{k,i}\phi_{k,i}) \\
\mu_{k,i+1} &= \beta\mu_{k,i} + \gamma(1-\beta)\left(\frac{1}{\delta + \frac{1}{L}u'^{*}_{k,i}u'_{k,i}}\right) \\
q_{k,i} &= u^{*}_{k,i}(d_k(i) - u_{k,i}\psi_{k,i+1}) \\
c_{kl}(i+1) &= \frac{\|\,\psi_{k,i+1} + \mu_{k,i+1}q_{k,i} - \psi_{l,i+1}\,\|^{-2}}{\sum_{j\in\mathcal{N}_k}\|\,\psi_{k,i+1} + \mu_{k,i+1}q_{k,i} - \psi_{j,i+1}\,\|^{-2}} \\
\phi_{k,i+1} &= \sum_{l\in\mathcal{N}_k} c_{kl}(i+1)\psi_{l,i+1}
\end{aligned} \tag{4.23}$$

The DCTLMS variable step-size algorithm was declared in (3.50) for single-task networks while the mean transient analysis and the mean-square performance have been stated in (4.18) and (4.19), respectively.

The variable step-size based on DCTLMS for multi-task networks with adaptive clustering technique is summarized in Algorithm B.4 (A) (See Appendix-B).

The theoretical derivations of the variable step-size algorithm based on DCTLMS with adaptive combiners that considers the aforementioned equations for mean transient and mean-square analyses for multi-task networks are outlined in Algorithm B.4 (B) (See Appendix-B).

4.4.4 Transform Domain Variable Step-Size DLMS (TD-VSS) with Adaptive Combiners

The fourth algorithm was derived in Chapter 3 for single-task networks [9] which can be treated as multi-task networks by employing adaptive clustering as discussed in earlier section 4.1. The weight-update formulas for single-task networks were

$$
\begin{aligned}
\psi_{k,i+1} &= \phi_{k,i} + \frac{\mu_k(i)}{\delta + \sigma^2_{k,i}} u^*_{k,i}(\, d_k(i) - u_{k,i}\phi_{k,i}) \\
\sigma^2_{k,i+1} &= \beta\sigma^2_{k,i} + (\, 1 - \beta)\, |u_{k,i}|^2 \\
A_k(i) &= \alpha\mu_k(i) + \frac{\gamma}{L} \sum_{j=i-(\, L-1)}^{i} e^2_k(j) \\
\mu_k(i+1) &= \begin{cases} A_k(i), & \text{if } i = kL \text{ and } A_k(i) \in (\, \mu_{k,min}, \mu_{k,max}) \\ \mu_{k,max}, & \text{if } i = kL \text{ and } A_k(i) \geq \mu_{k,max} \\ \mu_{k,min}, & \text{if } i = kL \text{ and } A_k(i) \leq \mu_{k,min} \\ \mu_k(i), & \text{if } i \neq kL \end{cases} \\
\phi_{k,i+1} &= \sum_{l \in \mathcal{N}_k} c_{kl}\psi_{l,i+1}
\end{aligned}
\tag{4.24}
$$

In multi-task networks, the weight-update formulas become

$$
\begin{aligned}
\psi_{k,i+1} &= \phi_{k,i} + \frac{\mu_k(i)}{\delta + \sigma^2_{k,i}} u^*_{k,i}(\, d_k(i) - u_{k,i}\phi_{k,i}) \\
\sigma^2_{k,i+1} &= \beta\sigma^2_{k,i} + (\, 1 - \beta)\, |u_{k,i}|^2
\end{aligned}
$$

$$A_k(i) = \alpha\mu_k(i) + \frac{\gamma}{L}\sum_{j=i-(L-1)}^{i} e_k^2(j)$$

$$\mu_k(i+1) = \begin{cases} A_k(i), & \text{if } i = kL \text{ and } A_k(i) \in (\,\mu_{k,min}, \mu_{k,max}) \\ \mu_{k,max}, & \text{if } i = kL \text{ and } A_k(i) \geq \mu_{k,max} \\ \mu_{k,min}, & \text{if } i = kL \text{ and } A_k(i) \leq \mu_{k,min} \\ \mu_k(i), & \text{if } i \neq kL \end{cases}$$

$$q_{k,i} = u_{k,i}^*(\,d_k(i) - u_{k,i}\psi_{k,i+1})$$

$$c_{kl}(i+1) = \frac{\parallel \psi_{k,i+1} + \mu_k(i+1)q_{k,i} - \psi_{l,i+1} \parallel^{-2}}{\sum_{j\in\mathcal{N}_k} \parallel \psi_{k,i+1} + \mu_k(i+1)q_{k,i} - \psi_{j,i+1} \parallel^{-2}}$$

$$\phi_{k,i+1} = \sum_{l\in\mathcal{N}_k} c_{kl}(i+1)\psi_{l,i+1} \tag{4.25}$$

The transform domain variable step-size DLMS (TDVSS) for Multi-Task Networks with adaptive combiners is summarized in Algorithm B.5 (A) (See Appendix-B). The theoretical derivations of the transform domain variable step-size algorithm that consider the aforementioned equations for mean transient and mean-square analysis for multi-task networks using TDVSS with adaptive combiners are outlined in Algorithm B.5 (B) (See Appendix-B).

4.4.5 New Variable Step-Size Transform-Domain DLMS (NVSTDLMS) with Adaptive Combiners

The fifth algorithm of VSSTDMDLMS was discussed in Chapter 3 for single-task networks. The weight-update formulas for single-task networks were

$$
\begin{aligned}
\psi_{k,i+1} &= \phi_{k,i} + \frac{\mu_k(i)}{\delta + \sigma^2_{k,i}} u^*_{k,i}(\, d_k(i) - u_{k,i}\phi_{k,i}) \\
\sigma^2_{k,i+1} &= \beta\sigma^2_{k,i} + (\,1-\beta)\,|u_{k,i}|^2 \\
\mu_k(i+1) &= \begin{cases} \frac{\varepsilon_{1,k}(i)}{\varepsilon_{2,k}(i)}, & \text{if } \mu_k(i+1) \in (\,\mu_{k,min}, \mu_{k,max}) \\ \mu_{k,max}, & \text{if } \mu_k(i+1) \geq \mu_{k,max} \\ \mu_{k,min}, & \text{if } \mu_k(i+1) \leq \mu_{k,min} \end{cases} \\
\phi_{k,i+1} &= \sum_{l \in \mathcal{N}_k} c_{kl}\psi_{l,i+1}
\end{aligned} \tag{4.26}
$$

In the sequel, the adaptive combiner in Section 4.1 is included to achieve our proposed multi-task networks. The weight-update formulas for multi-task networks become

$$
\begin{aligned}
\psi_{k,i+1} &= \phi_{k,i} + \frac{\mu_k(i)}{\delta + \sigma^2_{k,i}} u^*_{k,i}(\, d_k(i) - u_{k,i}\phi_{k,i}) \\
\sigma^2_{k,i+1} &= \beta\sigma^2_{k,i} + (\,1-\beta)\,|u_{k,i}|^2
\end{aligned}
$$

$$\mu_k(i+1) = \begin{cases} \frac{\varepsilon_{1,k}(i)}{\varepsilon_{2,k}(i)}, & \text{if } \mu_k(i+1) \in (\,\mu_{k,min}, \mu_{k,max}) \\ \mu_{k,max}, & \text{if } \mu_k(i+1) \geq \mu_{k,max} \\ \mu_{k,min}, & \text{if } \mu_k(i+1) \leq \mu_{k,min} \end{cases}$$

$$\begin{aligned} q_{k,i} &= u_{k,i}^{*}(\,d_k(i) - u_{k,i}\psi_{k,i+1}) \\ c_{kl}(i+1) &= \frac{\parallel \psi_{k,i+1} + \mu_k(i+1)q_{k,i} - \psi_{l,i+1} \parallel^{-2}}{\sum_{j\in\mathcal{N}_k} \parallel \psi_{k,i+1} + \mu_k(i+1)q_{k,i} - \psi_{j,i+1} \parallel^{-2}} \\ \phi_{k,i+1} &= \sum_{l\in\mathcal{N}_k} c_{kl}(i+1)\psi_{l,i+1} \end{aligned} \tag{4.27}$$

The variable step-size DLMS algorithm based on NVSTDLMS for Multi-Task Networks with adaptive combiners is summarized in Algorithm B.6 (A) (See Appendix-B).

The theoretical derivations of the variable step-size algorithm based on NVST-DLMS with adaptive networks that consider the aforementioned equations for mean transient and mean-square analyses for multi-task networks are outlined in Algorithm B.6 (B) (See Appendix-B).

4.4.6 Optimal Step-Size Transform Domain DLMS (VSSTDLMS) with Adaptive Combiners

The sixth algorithm was derived in Chapter 3 for single-task networks [10] that can be converted to multi-task networks when adaptive clustering is considered as discussed earlier in Section 4.1. In single-task networks, the weight-update formulas were

$$
\begin{aligned}
\psi_{k,i+1} &= \phi_{k,i} + \frac{\mu_k(i)}{\delta + \sigma_{k,i}^2} u_{k,i}^* (\, d_k(i) - u_{k,i}\phi_{k,i}) \\
\sigma_{k,i+1}^2 &= \beta\sigma_{k,i}^2 + (\, 1-\beta)\, |u_{k,i}|^2 \\
\rho_k(i+1) &= \gamma\rho_k(i) + (\, 1-\gamma)\, \left(e_k^2(i)\hat{u}_{k,i}^T\hat{u}_{k,i}\right) \\
\eta_k(i+1) &= \gamma\eta_k(i) + (\, 1-\gamma)\, \big(e_k(i)e_k(i-1)\big) \\
\mu_k(i+1) &= \frac{\big|\eta_k(i+1)\big|}{\rho_k(i+1)} \\
\phi_{k,i+1} &= \sum_{l\in\mathcal{N}_k} c_{kl}\psi_{l,i+1}
\end{aligned} \tag{4.28}
$$

The weight-update formulas for multi-task networks become

$$
\begin{aligned}
\psi_{k,i+1} &= \phi_{k,i} + \frac{\mu_k(i)}{\delta + \sigma_{k,i}^2} u_{k,i}^* (\, d_k(i) - u_{k,i}\phi_{k,i}) \\
\sigma_{k,i+1}^2 &= \beta\sigma_{k,i}^2 + (\, 1-\beta)\, |u_{k,i}|^2 \\
\rho_k(i+1) &= \gamma\rho_k(i) + (\, 1-\gamma)\, \left(e_k^2(i)\hat{u}_{k,i}^T\hat{u}_{k,i}\right) \\
\eta_k(i+1) &= \gamma\eta_k(i) + (\, 1-\gamma)\, \big(e_k(i)e_k(i-1)\big) \\
\mu_k(i+1) &= \frac{\big|\eta_k(i+1)\big|}{\rho_k(i+1)} \\
q_{k,i} &= u_{k,i}^*(\, d_k(i) - u_{k,i}\psi_{k,i+1}) \\
c_{kl}(i+1) &= \frac{\|\, \psi_{k,i+1} + \mu_k(i+1)q_{k,i} - \psi_{l,i+1}\, \|^{-2}}{\sum_{j\in\mathcal{N}_k} \|\, \psi_{k,i+1} + \mu_k(i+1)q_{k,i} - \psi_{j,i+1}\, \|^{-2}} \\
\phi_{k,i+1} &= \sum_{l\in\mathcal{N}_k} c_{kl}(i+1)\psi_{l,i+1}
\end{aligned} \tag{4.29}
$$

where $\hat{u}_{k,i} = V_{k,i}^{-1} u_{k,i}$ with $V_{k,i}^2 = \text{diag}\{\sigma_{1,k,i}^2, \sigma_{2,k,i}^2, \ldots, \sigma_{M,k,i}^2\}$.

The optimal variable step-size transform domain DLMS (VSSTDLMS) for multi-task networks with adaptive combiners is summarized in Algorithm B.7 (A) (See Appendix-B).

The theoretical derivations of the optimal variable step-size algorithm that consider the aforementioned equations for mean transient and mean-square analyses for multi-task Networks with adaptive combiners are outlined in Algorithm B.7 (B) (See Appendix-B).

4.5 Multi-Task Networks Simulations

In what follows, we discuss several experiments to validate the theoretical derivations for the multi-task networks algorithms discussed in the previous sections. For all the simulations, the input regressors are represented as

$$u_{k,i} = \text{col}\{u_k(i), u_k(i-1), \ldots, u_k(i-(M-1))\} \tag{4.30}$$

with zero-mean and variance of one ($\sigma_{u,k}^2 = 1$).

In our experiments, 100 independent simulations were conducted and averaged. The transient plots were produced by executing the multi-task networks training task for less than 6000 iterations. The desired data $d_k(i)$ was produced according to the model in (3.8) with the unknown multi-task filter weights $w_1^o = \text{col}\{1, 1\}$, $w_2^o = \text{col}\{0.5, 0.5\}$, and $w_3^o = \text{col}\{0.7, 0.7\}$ of length $(M \times 1)$ with filter length of

$M = 2$ taps. Figure 4.2 represents the topology of multi-task networks with three clusters. The noise variance for the AWGN background noise of the multi-task networks is shown in Figure 4.3.

The experimental parameters (i.e., standard LMS, traditional TDLMS [5], DCTLMS [8], TDVSS [9], VSSTDLMS [10], and NVSTDLMS [11], VSSLMS [3]) are similar to the setup in [9, 11], these are in Table 4.1.

The performance of multi-task networks algorithms for two cases, namely: single node with adaptive combiners and collaborative nodes with adaptive combiners are shown in Figures 4.4, 4.5, 4.6, 4.7, 4.8, and 4.9. In each case, the experiments considered theoretical expressions and practical simulations with a good match between them. The setup of the experiments was selected carefully to carry out a fair comparison among the algorithms.

The best performance was for the VSSLMS and DCTLMS algorithms with a MSD of -57 dB and -55 dB, respectively, when the nodes collaborate together with adaptive combination. In the case of a single node with adaptive combiners, the MSD degraded to -53.5 dB and -51.5 dB, respectively. The gain in the case of collaborative nodes with adaptive combiners over a single node was -3.5 dB.

The performance of DLMS and TDVSS algorithms was identical with a MSD of -47.5 dB for the case of collaborative nodes with adaptive combiners and it degraded to -44 dB in the case of a single node with adaptive combiners. In these experiments, the gain of the collaborative nodes with adaptive combiners was constant at -3.5 dB.

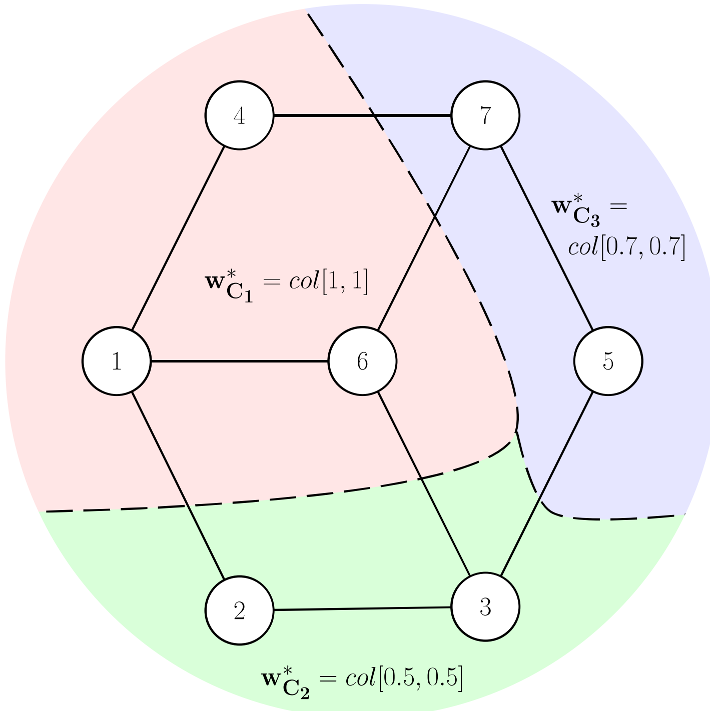

Figure 4.2: A Multi-Task Network Topology with $N = 7$ Nodes. The optimum filters are $w_1^o = \text{col}\{1, 1\}$, $w_2^o = \text{col}\{0.5, 0.5\}$, and $w_3^o = \text{col}\{0.7, 0.7\}$.

The worst performance was for TDLMS and NVSTDLMS algorithms with an identical MSD of -46 dB for the case of collaborative nodes with adaptive combiners and it degraded to -43.5 dB in the case of a single node with adaptive combiners. In these experiments, the gain of using collaborative nodes with adaptive combiners was constant at -2.5 dB.

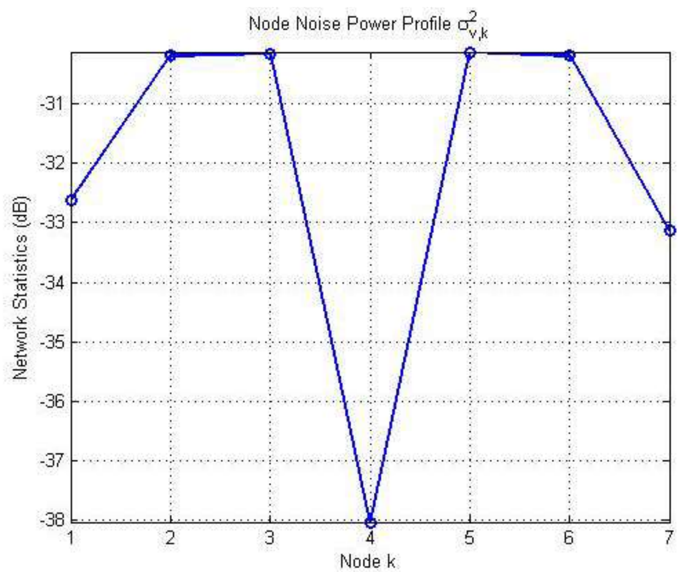

Figure 4.3: Noise Power Profile.

Table 4.1: Parameter Settings For Multi-Task Networks Algorithms

DLMS	**NVSTDLMS**
$\mu = 5 \times 10^{-2}$	$\mu_{max} = 1 \times 10^{-1}$ $\alpha_1 = 1 - 7 \times 10^{-3}$ $\mu_{min} = 5 \times 10^{-2}$ $\alpha_2 = 1 - 10^{-5}$ $\beta = 0.9$ $\delta = 2.5 \times 10^{-2}$
TDLMS	**VSSTDLMS**
$\mu = 5 \times 10^{-2}$ $\beta = 0.9$ $\delta = 2.5 \times 10^{-2}$	$\gamma = 0.98$ $\beta = 0.95$ $\delta = 2.5 \times 10^{-2}$
VSSLMS	**DCTLMS**
$\mu = 5 \times 10^{-2}$ $\alpha = 0.9985$ $\gamma = 8 \times 10^{-3}$	$\gamma = 8 \times 10^{-3}$ $\beta = 0.9985$ $\delta = 8 \times 10^{-4}$ $L = 10$
TDVSS	
$\mu_{max} = 1 \times 10^{-1}$ $\alpha = 0.9985$ $\beta = 0.9$ $L = 10$ $\mu_{min} = 5 \times 10^{-2}$ $\gamma = 8 \times 10^{-3}$ $\delta = 2.5 \times 10^{-2}$	

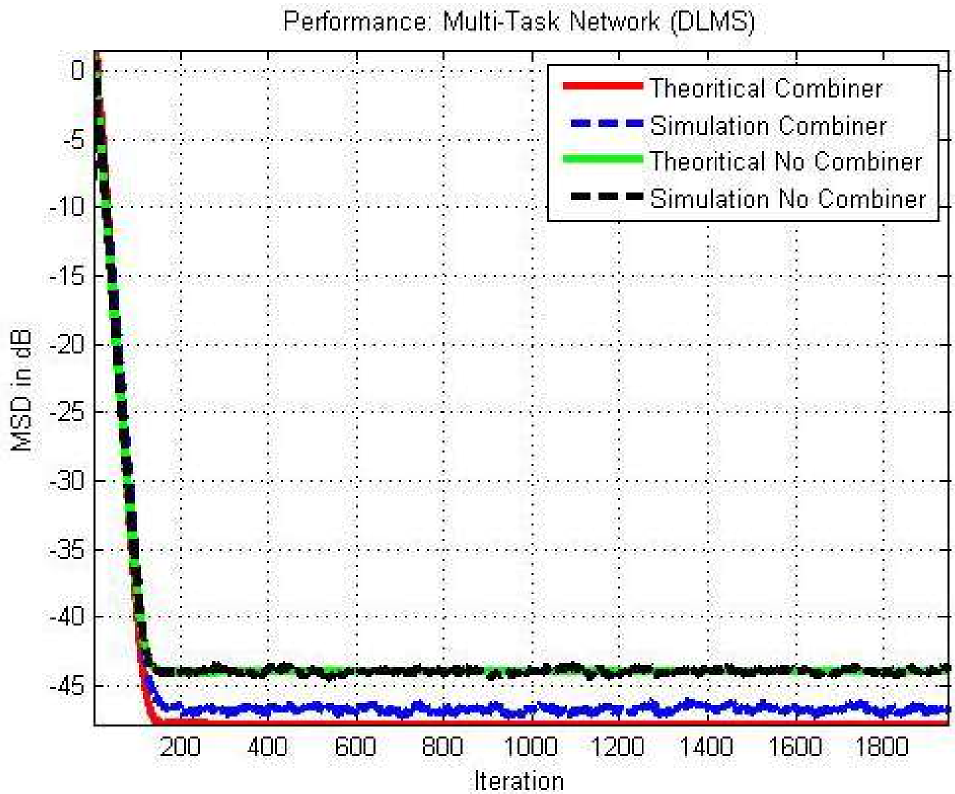

Figure 4.4: Performance of Multi-Task Networks using DLMS.

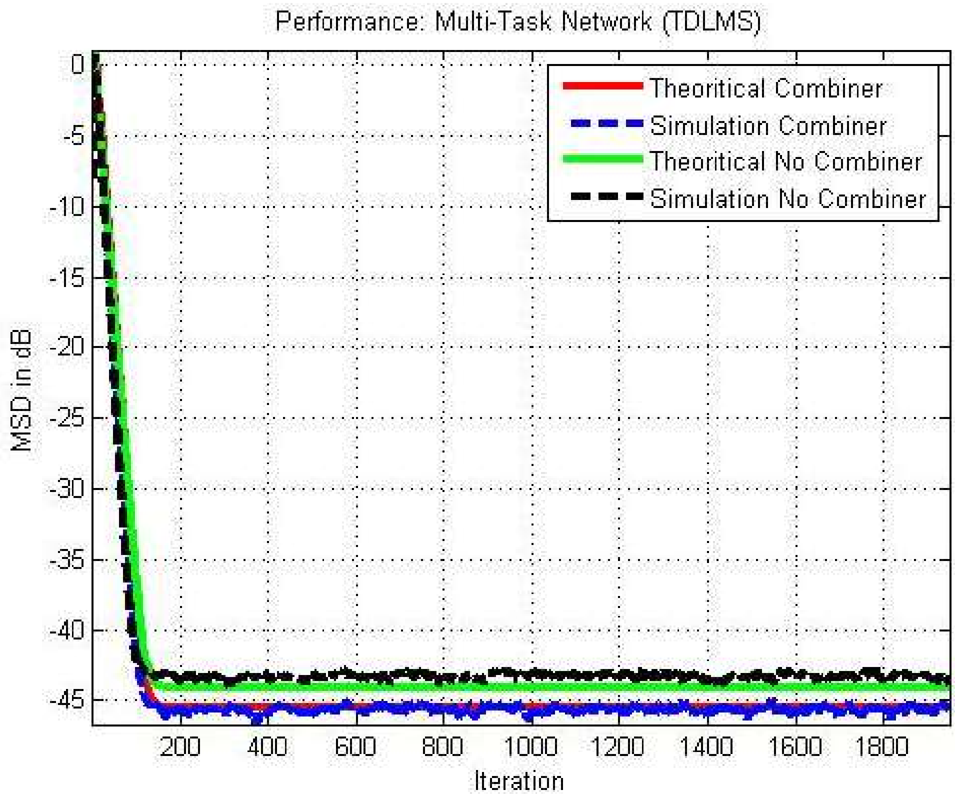

Figure 4.5: Performance of Multi-Task Networks using TDLMS.

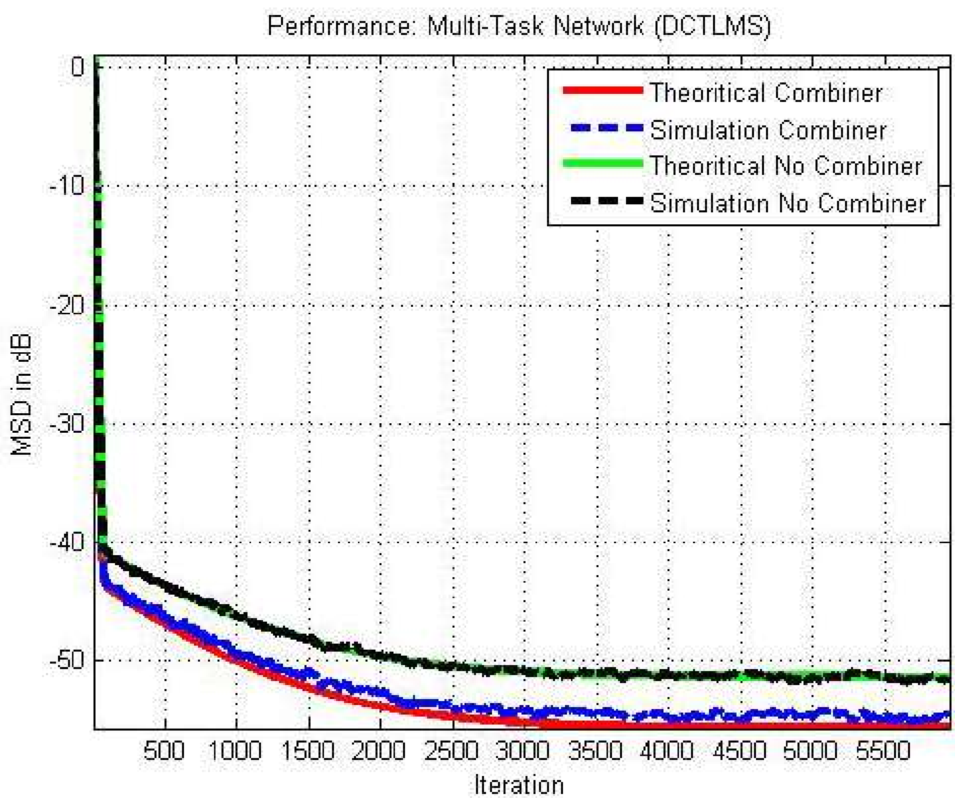

Figure 4.6: Performance of Multi-Task Networks using DCTLMS.

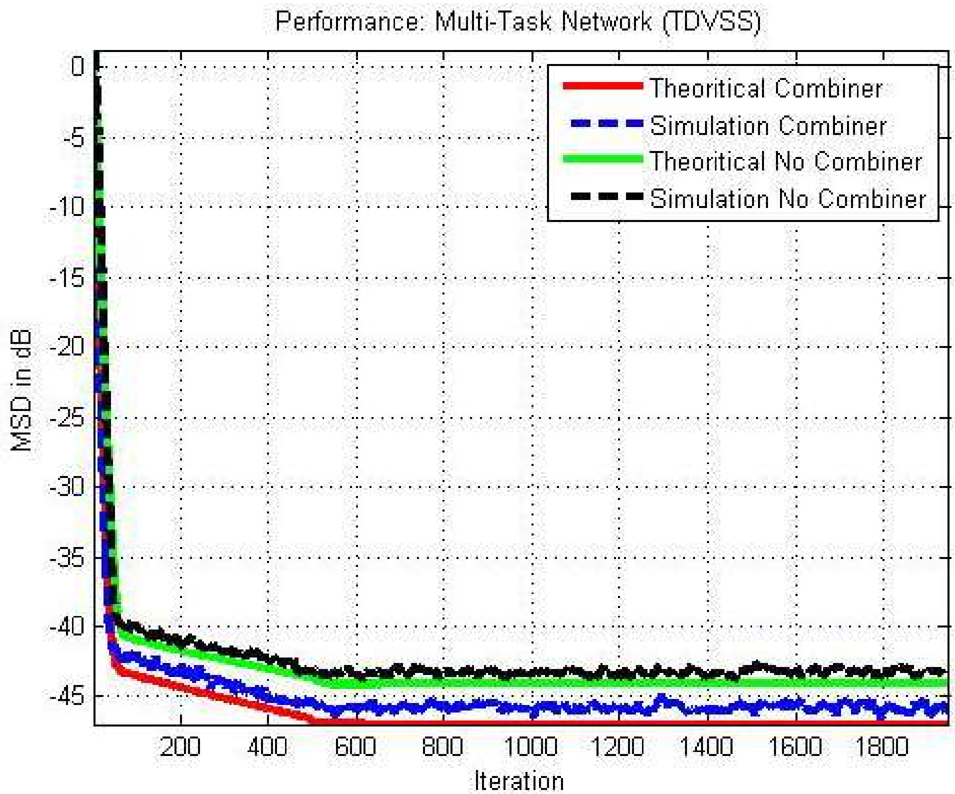

Figure 4.7: Performance of Multi-Task Networks using TDVSS.

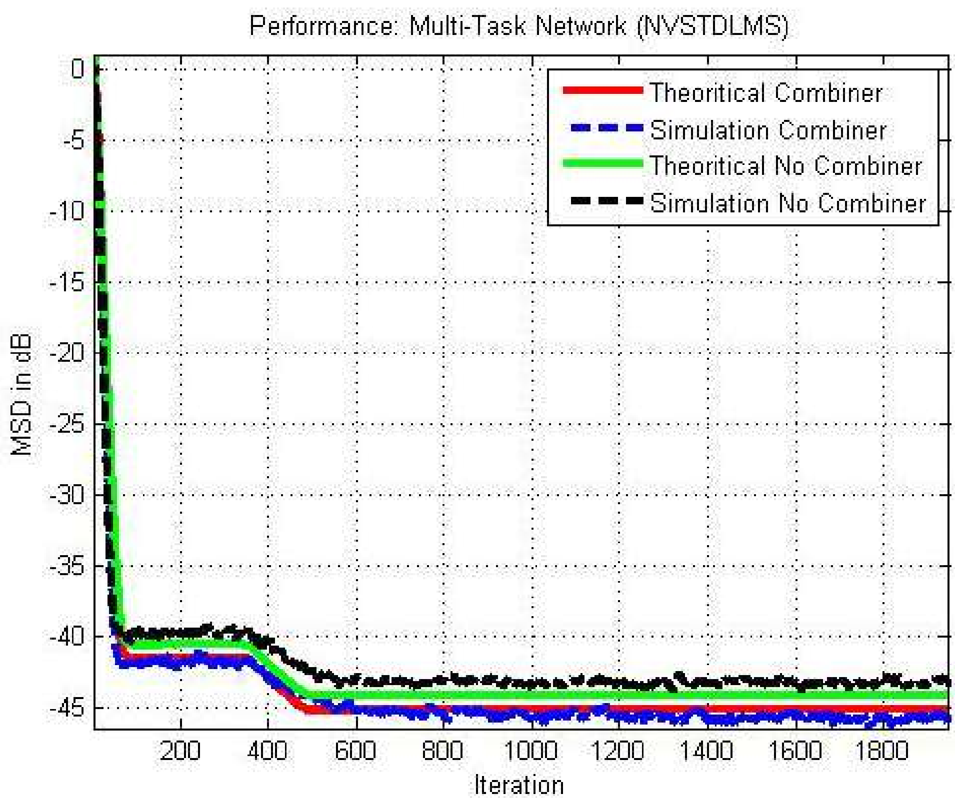

Figure 4.8: Performance of Multi-Task Networks using NVSTDLMS.

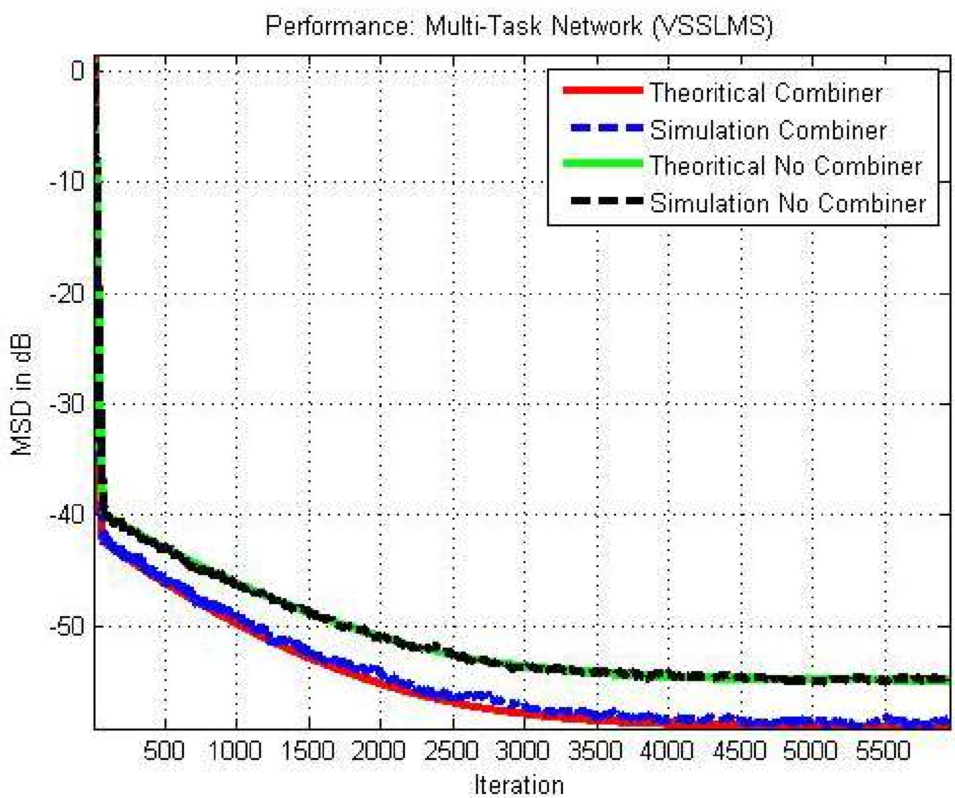

Figure 4.9: Performance of Multi-Task Networks using VSSLMS.

4.6 Discussion of The Results

In multi-task networks, the adaptive combiner matrix has been selected to grant the right stochastic property which yield the derivation of the theoretical equations of the transform domain algorithms. Before that, the formulation of variable step-size transform domain DLMS algorithms for multi-task networks was provided. The mismatch between theoretical and practical simulation results increased as the two essential parameters have larger values. Indeed, these parameters were step-size and AWGN values which is a very trivial obstacle and they should be kept small to have better performance results. There is a tradeoff between the time taken for MSD to converge and the selection of step-size and AWGN level. Indeed, when the combination has small value, the convergence iteration lasts longer and vice versa.

Again, the DCTDLMS and the VSSLMS algorithms of multi-task networks have outperformed the rest of the algorithms. They converged around -54 dB and -58 dB, respectively, as shown in Figure 4.10 while the other algorithms converged at -47 dB. As the convergence factor α gets reduced from 8×10^{-3} to 1×10^{-3}, the MSD performance of DCTLMS and VSSLMS algorithms has been improved further to -64 dB and -68 dB, respectively, as illustrated in Figure 4.11 and Figure 4.12. It has been shown that it can be further improved by tuning two parameters, namely: α and β to lower values.

Also, the performance of TDLMS, TDVSS, NVSTDLMS, and VSSTDLMS algorithms for multi-task networks had similar results in comparison to the DLMS

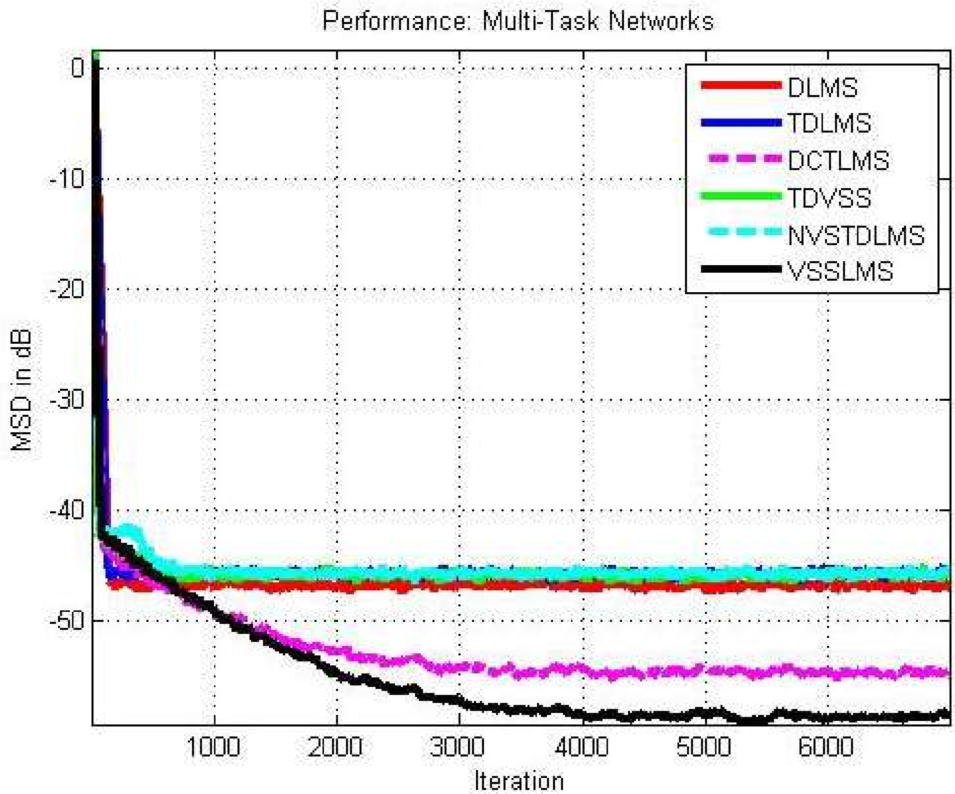

Figure 4.10: Performance Comparison of All Multi-Task Networks Algorithms.

multi-task networks. This issue can be addressed by readjusting the parameter settings to achieve better performance. Indeed, we intended to keep the parameter settings in a manner that they behaved similarly for fair comparison, but as expected the DCLMS and the VSSLMS algorithms of multi-task networks produced the best results.

4.7 Summary

In this chapter, we have developed many variable step-size transform domain algorithms for multi-task networks. The adaptive combiners have been used to convert the aforementioned variable step-size algorithms for single-task networks that dis-

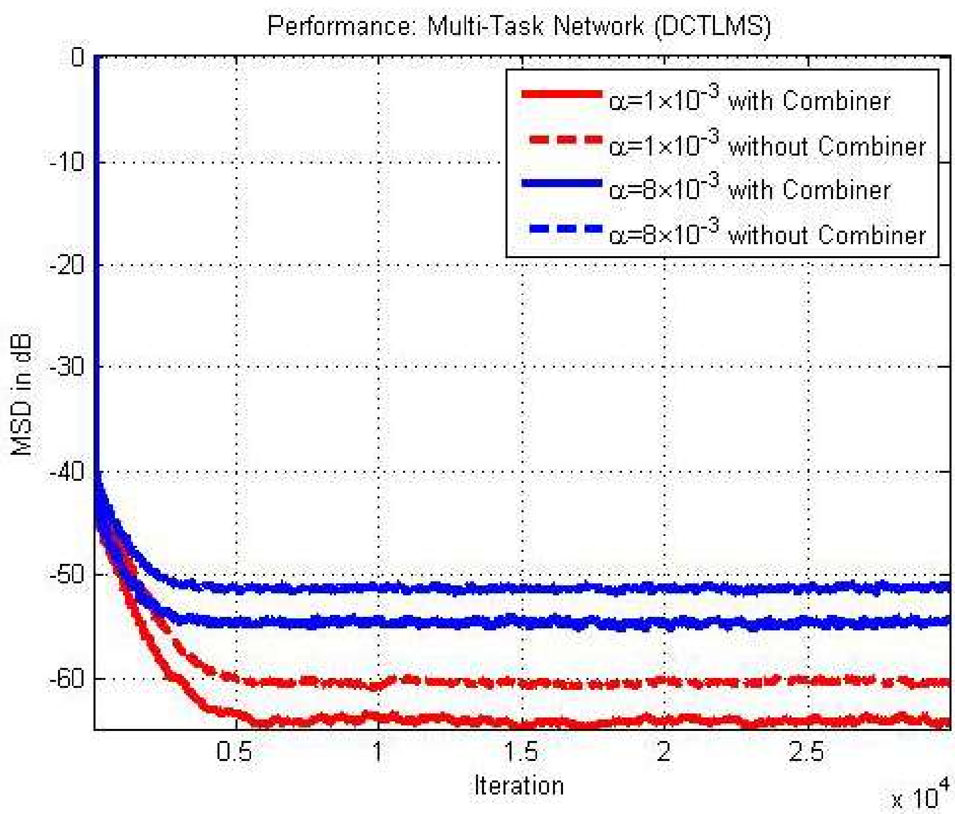

Figure 4.11: Performance Comparison of Multi-Task Networks using DCTLMS when α Get Replaced By Lower Value where Other Settings Preserved Fixed.

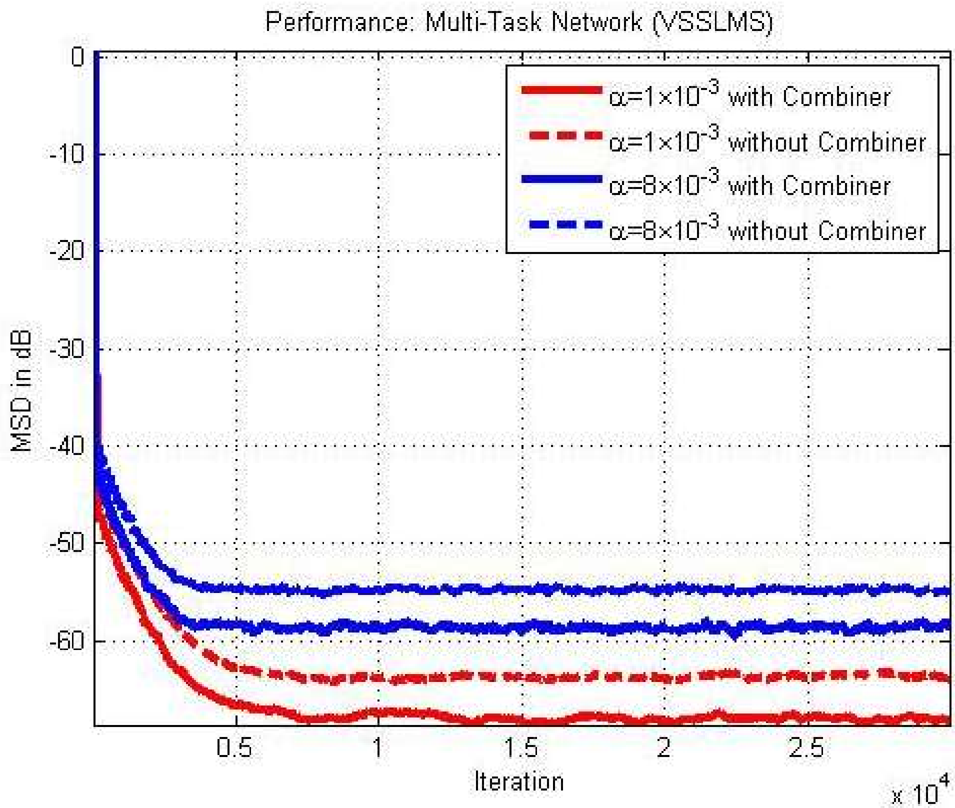

Figure 4.12: Performance Comparison of Multi-Task Networks using VSSLMS DLMS when α Get Replaced By Lower Value where Other Settings Preserved Fixed.

cussed in previous chapter to multi-task networks. The theoretical equations such as mean analysis and mean-square of the wight error have been derived as well as the mean of adaptive combiners. Our simulations have shown that VSSDLMS and DCTLMS algorithms outperform the conventional DLMS.

CHAPTER 5

CONCLUSION AND FUTURE RESEARCH WORK

5.1 Conclusion

In what follows, we summarize the main contributions made under the different chapters of the thesis.

In chapter 1, some basic definitions of single-task, multi-task, clustered multi-task networks have been outlined. The importance of channel estimation has been discussed to mitigate the deficiency of equalizers. We mention that the thesis considers only one type of adaptive filters which is LMS. The chapter outlines the major contributions under two scenarios, namely: single-task and multi-task networks.

In chapter 2, adaptive filter algorithms for single node networks have been discussed, namely: TDLMS, DCTLMS, TDVSS, NVSTDLMS, VSSTDLMS, and

VSSLMS algorithms. The latest literature review was discussed with emphasis on transform domain adaptive filtering.

In chapter 3, the estimation problem has been formulated for single-task network. Numerous variants of the variable step-size transform domain LMS in the case of single-node networks have been extended to single-task networks. We have formulated and analyzed successfully the different proposed algorithms. Extensive simulations have been performed to validate the theoretical derivations which matched well with the simulations. Typically, the MSD performance has been derived and shown to correctly match the simulation results. The best performance has been achieved with the VSSLMS and DCTLMS algorithms.

On the other hand, the multi-task networks have been formulated in chapter 4 by extending the existing work of single-task networks with the assistance of adaptive combiners. The performance of multi-task networks have been further enhanced by formulating different variants of the step-size transform domain algorithms of traditional single-node LMS Networks. The VSSLMS and DCTLMS techniques have again shown superior MSD performance in contrast to other algorithms.

The parameter setting (i.e., α) for DCTLMS and VSSLMS algorithms has been tested for two scenarios, typically: large and small values. It has been shown in the discussion that the performance improves when α takes smaller values in the case of DCTLMS as well as for VSSLMS. We show that different performance results can be achieved when parameter setting is readjusted.

Chapter 5 concludes the thesis with some future research directions.

5.2 Future Research Directions

In future work, the variable step-size transform domain algorithms DLMS for single and multi task networks can be extended to different filters. The possible research extensions include:

1. Incremental LMS strategies (ILMS): For the case of single-task networks, our approach should perform well. But, our approach will not work correctly for the case of multi-task networks due to the absence of adaptive combiners. In fact, the ILMS strategies don't permit to select which neighbours to aggregate. Since, multi-task networks for DLMS strategies have been derived based on adaptive combiners in this thesis.

2. Recursive Least Square (RLS): It is applicable for single and multi task networks. In case of multi-task networks, we claim that out approach should work for RLS. Since, RLS is an adaptive filter that can be distributed all over the network and it can be combined in order to improve its robustness gainst link failure as well as its performance.

3. Kalman Filters (KF): We believe that the kalman filters can be deployed in a distributed fashion. For this reason, we can focus on extending the algorithm in order to consider multi-task networks.

4. Adaptive combiners: Different adaptive combiners can be considered to achieve better performance in the case of Multi-Task Networks. In literature, there are many adaptive combiners that could be extended to this

work.

Appendix-A Single-Task Networks Algorithms

The list of theoretical and simulation procedures of the following algorithms are summarized in Appendix-A. The algorithms are

1. ATC DLMS
2. VSSLMS
3. TDLMS
4. DCTLMS
5. TDVSS
6. NVSTDLMS
7. VSSTDLMS

It is a useful practice when any single-task networks algorithm is simulated.

A.1 Adapt-Then-Combine DLMS

The Adapt-Then-Combine (ATC) single-task DLMS procedure is summarized in Algorithm A.1 (A) for easy simulation purposes.

Algorithm A.1 (A): ATC Diffusion LMS for Single-Task Networks

---

Initialize: $\phi_{k,0} = 0$ for all $k = 1, ..., N$.

Procedure: Start "for loop" at iteration $i \geq 1$ and node $1 \leq k \leq N$

Adapt Step:

$$\psi_{k,i+1} = \phi_{k,i} + \mu_k u^*_{k,i}(\, d_k(i) - u_{k,i}\phi_{k,i}) \tag{A.1}$$

Combine Step:

$$\phi_{k,i+1} = \sum_{l \in \mathcal{N}_k} c_{kl}\psi_{l,i+1} \tag{A.2}$$

The theoretical derivation that considers the aforementioned equations of mean transient and mean-square analyses in order to formulate single-task networks adaptive filter (DLMS) is short listed in Algorithm A.1 (B) for easy reference.

Algorithm A.1 (B): Theoretical MSD of ATC Diffusion LMS for Single-Task Networks

Initialize: $\eta(0) = \| \bar{w}^{(o)} \|^2$ and $\bar{F}^0 = I_{N^2M^2}$.

$$\bar{F} = (\, \bar{G}^*_{i-1} \odot \bar{G}^{*T}_{i-1})\, [\, I_{N^2M^2} - (\, I_{NM} \odot \Lambda D) \\ - (\, \Lambda D \odot I_{NM}) + (\, D \odot D)\mathcal{A}] \quad (N^2M^2 \times N^2M^2) \tag{A.3}$$

Procedure: Start "for loop" at iteration $i \geq 1$ and node $1 \leq k \leq N$

Global MSD:

$$N(i) = \eta(i-1) + b^T \bar{F}^i q_\eta - \| \bar{w}^{(o)} \|^2_{\bar{F}^i \left(I - \bar{F}\right) q_\eta} \tag{A.4}$$

A.2 Variable Step-Size DLMS (VSSLMS)

The variable step-size based on variable step-size DLMS (VSSLMS) procedure for single-task networks is summarized in Algorithm A.2 (A) for easy simulation purposes.

Algorithm A.2 (A): ATC Diffusion LMS for Variable Step-Size DLMS (VSSLMS) Single-Task Networks Algorithm

Initialize: $\phi_{k,0} = 0$ for all $k = 1, ..., N$.

Procedure: Start "for loop" at iteration $i \geq 1$ and node $1 \leq k \leq N$

Adapt Step:

$$\psi_{k,i+1} = \phi_{k,i} + \mu_k(i)u^*_{k,i}(\, d_k(i) - u_{k,i}\phi_{k,i}) \tag{A.5}$$

Update Step-Size:

$$\mu_k(i+1) = \alpha\mu_k(i) + \gamma e^2_k(i) \tag{A.6}$$

Combine Step:

$$\phi_{k,i+1} = \sum_{l \in \mathcal{N}_k} c_{kl}\psi_{l,i+1} \tag{A.7}$$

The theoretical derivation of VSSLMS considers the aforementioned equations for mean transient and mean-square analyses in order to formulate the single-task networks. It is short listed in Algorithm A.2 (B) for easy reference.

Algorithm A.2 (B): Theoretical MSD of ATC Diffusion LMS for Variable Step-Size (VSSLMS) Single-Task Networks.

Initialize: $\eta(0) = \parallel \bar{w}^{(o)} \parallel^2$ and $\bar{F}^0 = I_{N^2M^2}$.

$$\bar{F} = \left(\bar{G}^* \odot \bar{G}^{*T}\right) [\, I_{N^2M^2} - (\, I_{NM} \odot \Lambda E\left[\boldsymbol{D}_{i-1}\right])$$
$$- (\, \Lambda E\left[\boldsymbol{D}_{i-1}\right] \odot I_{NM}) + E\left[\boldsymbol{D}_{i-1} \odot \boldsymbol{D}_{i-1}\right] \mathcal{A}] \quad (N^2M^2 \times N^2M^2) \tag{A.8}$$

Procedure: Start "for loop" at iteration $i \geq 1$ and node $1 \leq k \leq N$

Update Step-Size:

$$E\left[\boldsymbol{D}_i^2\right] = \alpha^2 E\left[\boldsymbol{D}_{i-1}^2\right] + 2\alpha\gamma E\left[\boldsymbol{D}_{i-1}\right] E\left[\boldsymbol{E}^{i-1}\right] + \gamma^2 E\left[\boldsymbol{E}^{2,i-1}\right]$$
$$E\left[\boldsymbol{D}_i\right] = \alpha E\left[\boldsymbol{D}_{i-1}\right] + \gamma E\left[\boldsymbol{E}^{i-1}\right] \tag{A.9}$$

Global MSD:

$$N(i) = \eta(i-1) + b^T \bar{F}^i q_\eta - \parallel \bar{w}^{(o)} \parallel^2_{\bar{F}^i\left(I-\bar{F}\right)q_\eta} \tag{A.10}$$

A.3 Transform Domain DLMS (TDLMS)

The variable step-size transform domain DLMS (TDLMS) procedure for single-task networks is summarized in Algorithm A.3 (A) for easy simulation purposes.

Algorithm A.3 (A): ATC Diffusion LMS for Variable Step-

Size Transform-Domain (TDLMS) Single-Task Networks Algorithm.

Initialize: $\phi_{k,0} = 0$ for all $k = 1, ..., N$ and $u_{k,i}$ in transform domain.

Procedure: Start "for loop" at iteration $i \geq 1$ and node $1 \leq k \leq N$

Adapt Step:

$$\psi_{k,i+1} = \phi_{k,i} + \frac{\mu_k}{\delta + \sigma_{k,i}^2} u_{k,i}^* (\, d_k(i) - u_{k,i}\phi_{k,i}) \tag{A.11}$$

Update Power Estimate:

$$\sigma_{k,i+1}^2 = \beta\sigma_{k,i}^2 + (\,1 - \beta)\,|u_{k,i}|^2 \tag{A.12}$$

Combine Step:

$$\phi_{k,i+1} = \sum_{l \in \mathcal{N}_k} c_{kl}\psi_{l,i+1} \tag{A.13}$$

The theoretical derivation of variable step-size transform domain DLMS (TDLMS) that considers the aforementioned equations for mean transient and mean-square analyses in order to formulate single-task networks is short listed in Algorithm A.3 (B) for easy reference.

Algorithm A.3 (B): Theoretical MSD of ATC Diffusion LMS for Variable Step-Size Transform-Domain (TDLMS) Single-Task Networks.

Initialize: $\eta(0) = \| \bar{w}^{(o)} \|^2$ where $\bar{w}^{(o)}$ in transform domain and $\bar{F}^0 = I_{N^2M^2}$.

$$\begin{aligned} \bar{F} = &\left(\bar{G}^* \odot \bar{G}^{*T}\right) [\, I_{N^2M^2} - (\, I_{NM} \odot \Lambda E\, [\boldsymbol{D}_{i-1}]) \\ &- (\, \Lambda E\, [\boldsymbol{D}_{i-1}] \odot I_{NM}) \, + E\, [\boldsymbol{D}_{i-1} \odot \boldsymbol{D}_{i-1}]\, \mathcal{A}] \quad (N^2M^2 \times N^2M^2) \end{aligned} \tag{A.14}$$

Procedure: Start "for loop" at iteration $i \geq 1$ and node $1 \leq k \leq N$

Update Step-Size:

$$\begin{aligned} E\left[\boldsymbol{D}_i^2\right] &= diag\{E\left[\boldsymbol{\mu}_1^{'}(i)\right] I_M, E\left[\boldsymbol{\mu}_2^{'}(i)\right] I_M, ..., E\left[\boldsymbol{\mu}_N^{'}(i)\right] I_M\} \\ E\left[\boldsymbol{D}_i\right] &= diag\{E\left[\boldsymbol{\mu}_1^{'2}(i)\right] I_M, E\left[\boldsymbol{\mu}_2^{'2}(i)\right] I_M, ..., E\left[\boldsymbol{\mu}_N^{'2}(i)\right] I_M\} \end{aligned} \tag{A.15}$$

$$\begin{aligned} E\left[\boldsymbol{\mu}_k^{'}(i)\right] &= \mu_k \left[\frac{1}{E\left[\boldsymbol{\sigma}_{k,i}^2\right]} - \frac{\delta}{E\left[\boldsymbol{\sigma}_{k,i}^4\right]}\right] \\ E\left[\boldsymbol{\mu}_k^{'2}(i)\right] &= \mu_k^2 \left[\frac{1}{E\left[\boldsymbol{\sigma}_{k,i}^4\right]} + \frac{\delta^2}{E\left[\boldsymbol{\sigma}_{k,i}^8\right]} - \frac{2\delta}{E\left[\boldsymbol{\sigma}_{k,i}^6\right]}\right] \end{aligned} \tag{A.16}$$

$$\begin{aligned} E\left[\boldsymbol{\sigma}_{k,i}^2\right] &= \beta E\left[\boldsymbol{\sigma}_{k,i-1}^2\right] + (1-\beta)\,\sigma_k^2 \\ E\left[\boldsymbol{\sigma}_{k,i}^4\right] &= \beta^2 E\left[\boldsymbol{\sigma}_{k,i-1}^4\right] + 2\beta\,(1-\beta)\,\sigma_k^2 E\left[\boldsymbol{\sigma}_{k,i-1}^2\right] + (1-\beta)^2\,\sigma_k^4 \\ E\left[\boldsymbol{\sigma}_{k,i}^6\right] &= \beta^3 E\left[\boldsymbol{\sigma}_{k,i-1}^6\right] + 3\beta^2\,(1-\beta)\,\sigma_k^2 E\left[\boldsymbol{\sigma}_{k,i-1}^4\right] + 3\beta\,(1-\beta)^2\,\sigma_k^4 E\left[\boldsymbol{\sigma}_{k,i-1}^2\right] \end{aligned}$$

$$+ (1-\beta)^3 \sigma_k^6$$

$$E\left[\boldsymbol{\sigma}_{k,i}^8\right] = \beta^4 E\left[\boldsymbol{\sigma}_{k,i-1}^8\right] + 4\beta^3(1-\beta)\sigma_k^2 E\left[\boldsymbol{\sigma}_{k,i-1}^6\right] + 6\beta^2(1-\beta)^2\sigma_k^4 E\left[\boldsymbol{\sigma}_{k,i-1}^4\right]$$

$$+ 4\beta(1-\beta)^3\sigma_k^6 E\left[\boldsymbol{\sigma}_{k,i-1}^2\right] + (1-\beta)^4\sigma_k^8 \tag{A.17}$$

Global MSD:

$$\eta(i) = \eta(i-1) + b^T \bar{F}^i q_\eta - \| \bar{w}^{(o)} \|^2_{\bar{F}^i\left(I-\bar{F}\right)q_\eta} \tag{A.18}$$

A.4 Discrete Cosine Transform DLMS (DCTLMS)

The variable step-size based on DCTLMS procedure for single-task networks is summarized in Algorithm A.4 (A) for easy simulation purposes.

Algorithm A.4 (A): ATC Diffusion LMS for Variable Step-Size Single-Task Networks Based on Discrete Cosine Transform DLMS (DCTLMS) Algorithm

---

Initialize: $\phi_{k,0} = 0$ for all $k = 1, ..., N$ and $u_{k,i}$ in transform domain.

Procedure: Start "for loop" at iteration $i \geq 1$ and node $1 \leq k \leq N$

Adapt Step:

$$\psi_{k,i+1} = \phi_{k,i} + \mu_k(i) u_{k,i}^{*}(\, d_k(i) - u_{k,i}\phi_{k,i}) \tag{A.19}$$

Update Step-Size:

$$\mu_k(i+1) = \beta\mu_k(i) + \gamma(\, 1 - \beta)\left(\frac{1}{\delta + \frac{1}{L} u_{k,i}^{'*} u_{k,i}^{'}}\right) \tag{A.20}$$

Combine Step:

$$\phi_{k,i+1} = \sum_{l \in \mathcal{N}_k} c_{kl}\psi_{l,i+1} \tag{A.21}$$

The theoretical derivation of variable step-size based on DCTLMS that considers the aforementioned equations for mean transient and mean-square analyses in progress to formulate the single-task filter is short listed in Algorithm A.4 (B) for easy reference.

Algorithm A.4 (B): Theoretical MSD of ATC Diffusion LMS for Variable Step-Size Single-Task Networks Based on Discrete Cosine Transform (DCTLMS) Algorithm

---

Initialize: $\eta(0) = \parallel \bar{w}^{(o)} \parallel^2$ where $\bar{w}^{(o)}$ in transform domain and $\bar{F}^0 = I_{N^2M^2}$ for all $k = 1, ..., N$.

$$
\begin{aligned}
\bar{F} = &\left(\bar{G}^{*} \odot \bar{G}^{*T}\right)[\, I_{N^2M^2} - (\, I_{NM} \odot \Lambda E\,[\boldsymbol{D}_{i-1}]) \\
&- (\,\Lambda E\,[\boldsymbol{D}_{i-1}] \odot I_{NM}) \,+ E\,[\boldsymbol{D}_{i-1} \odot \boldsymbol{D}_{i-1}]\,\mathcal{A}] \quad (N^2M^2 \times N^2M^2)
\end{aligned} \tag{A.22}
$$

Procedure: Start "for loop" at iteration $i \geq 1$ and node $1 \leq k \leq N$

Update Step-Size:

$$
\begin{aligned}
E\left[\boldsymbol{\mu}_{k,i+1}\right] &= \beta E\left[\boldsymbol{\mu}_{k,i}\right] + \gamma\,(1-\beta)\left(\frac{L}{\sigma_k^2\,(L-2)}\right. \\
&\left. - \delta\frac{L^2}{\sigma_k^4\,(L-4)\,(L-2)}\right) \quad (M \times 1) \\
E\left[\boldsymbol{\mu}_{k,i+1}^2\right] &= \beta^2 E\left[\boldsymbol{\mu}_{k,i}^2\right] + 2\beta\gamma\,(1-\beta)\,E\left[\boldsymbol{\mu}_{k,i}\right]\left(\frac{L}{\sigma_k^2\,(L-2)}\right. \\
&\left. - \delta\frac{L^2}{\sigma_k^4\,(L-4)\,(L-2)}\right) + \gamma^2\,(1-\beta)^2\left(\frac{L^2}{\sigma_k^4\,(L-4)\,(L-2)}\right. \\
&- 2\delta\frac{L^3}{\sigma_k^6\,(L-6)\,(L-4)\,(L-2)} \\
&\left. + \delta^2\frac{L^4}{\sigma_k^8\,(L-8)\,(L-6)\,(L-4)\,(L-2)}\right) \quad (M \times 1)
\end{aligned} \tag{A.23}
$$

Global MSD:

$$
\eta(i) = \eta(i-1) + b^T \bar{F}^i q_\eta - \parallel \bar{w}^{(o)} \parallel^2_{\bar{F}^i\left(I-\bar{F}\right)q_\eta} \tag{A.24}
$$

A.5 Transform Domain Variable Step-Size DLMS (TDVSS)

The transform domain variable step-size DLMS (TDVSS) procedure for single-task networks is summarized in Algorithm A.5 (A) for easy simulation purposes.

Algorithm A.5 (A): ATC Diffusion LMS for Transform Domain Variable Step-Size (TDVSS) Single-Task Networks Algorithm

---

Initialize: $\phi_{k,0} = 0$ for all $k = 1, ..., N$ and $u_{k,i}$ in transform domain.

Procedure: Start "for loop" at iteration $i \geq 1$ and node $1 \leq k \leq N$

Adapt Step:

$$\psi_{k,i+1} = \phi_{k,i} + \frac{\mu_k(i)}{\delta + \sigma^2_{k,i}} u^*_{k,i}(\, d_k(i) - u_{k,i}\phi_{k,i}) \tag{A.25}$$

Update Power Estimate:

$$\sigma^2_{k,i+1} = \beta\sigma^2_{k,i} + (\,1 - \beta)\,|u_{k,i}|^2 \tag{A.26}$$

Update Step-Size:

$$A_k(i) = \alpha\mu_k(i) + \frac{\gamma}{L} \sum_{j=i-(\,L-1)}^{i} e^2_k(j)$$

$$\mu_k(i+1) = \begin{cases} A_k(i), & \text{if } i = kL \text{ and } A_k(i) \in (\,\mu_{k,min}, \mu_{k,max}) \\ \\ \mu_{k,max}, & \text{if } i = kL \text{ and } A_k(i) \geq \mu_{k,max} \\ \\ \mu_{k,min}, & \text{if } i = kL \text{ and } A_k(i) \leq \mu_{k,min} \\ \\ \mu_k(i), & \text{if } i \neq kL \end{cases} \tag{A.27}$$

Combine Step:

$$\phi_{k,i+1} = \sum_{l \in \mathcal{N}_k} c_{kl} \psi_{l,i+1} \tag{A.28}$$

The theoretical derivation of transform domain variable step-size DLMS (TDVSS) that considers the aforementioned equations for mean transient and mean-square analyses in order to formulate the single-task networks is short listed in Algorithm A.5 (B) for easy reference.

Algorithm A.5 (B): Theoretical MSD of ATC Diffusion LMS for Transform Domain Variable Step-Size (TDVSS) Single-Task Networks.

Initialize: $\eta(0) = \| \bar{w}^{(o)} \|^2$ where $\bar{w}^{(o)}$ in transform domain and $\bar{F}^0 = I_{N^2M^2}$.

$$\bar{F} = \left(\bar{G}^* \odot \bar{G}^{*T}\right) [\, I_{N^2M^2} - (\, I_{NM} \odot \Lambda E\, [\boldsymbol{D}_{i-1}])$$

$$-\left(\Lambda E\left[\boldsymbol{D}_{i-1}\right]\odot I_{NM}\right)+E\left[\boldsymbol{D}_{i-1}\odot\boldsymbol{D}_{i-1}\right]\mathcal{A}]\quad\left(N^2M^2\times N^2M^2\right)\tag{A.29}$$

Procedure: Start "for loop" at iteration $i \geq 1$ and node $1 \leq k \leq N$

Update Step-Size:

$$\begin{aligned}E\left[\boldsymbol{D}_i^2\right]&=diag\{E\left[\boldsymbol{\mu}_1^{'}(i)\right]I_M,E\left[\boldsymbol{\mu}_2^{'}(i)\right]I_M,...,E\left[\boldsymbol{\mu}_N^{'}(i)\right]I_M\}\\E\left[\boldsymbol{D}_i\right]&=diag\{E\left[\boldsymbol{\mu}_1^{'2}(i)\right]I_M,E\left[\boldsymbol{\mu}_2^{'2}(i)\right]I_M,...,E\left[\boldsymbol{\mu}_N^{'2}(i)\right]I_M\}\end{aligned}\tag{A.30}$$

$$\begin{aligned}E\left[\boldsymbol{\mu}_k^{'}(i)\right]&=E\left[\boldsymbol{\mu}_k(i)\right]\left[\frac{1}{E\left[\boldsymbol{\sigma}_{k,i}^2\right]}-\frac{\delta}{E\left[\boldsymbol{\sigma}_{k,i}^4\right]}\right]\\E\left[\boldsymbol{\mu}_k^{'2}(i)\right]&=E\left[\boldsymbol{\mu}_k^2(i)\right]\left[\frac{1}{E\left[\boldsymbol{\sigma}_{k,i}^4\right]}+\frac{\delta^2}{E\left[\boldsymbol{\sigma}_{k,i}^8\right]}-\frac{2\delta}{E\left[\boldsymbol{\sigma}_{k,i}^6\right]}\right]\end{aligned}\tag{A.31}$$

$$\begin{aligned}E\left[\boldsymbol{\mu}_k(i)\right]&=\alpha E\left[\boldsymbol{\mu}_k(i-1)\right]+\frac{\gamma}{L}\sum_{j=i-(L-1)}^{i}E\left[\boldsymbol{e}_k^2(j)\right]\\E\left[\boldsymbol{\mu}_k^2(i)\right]&=\alpha^2E\left[\boldsymbol{\mu}_k^2(i-1)\right]\\&\quad+2\alpha\frac{\gamma}{L}\sum_{j=i-(L-1)}^{i}E\left[\boldsymbol{e}_k^2(j)\right]E\left[\boldsymbol{\mu}_k(i-1)\right]\\&\quad+\frac{\gamma^2}{L^2}\sum_{j=i-(L-1)}^{i}E\left[\boldsymbol{e}_k^4(j)\right]\end{aligned}\tag{A.32}$$

$$E\left[\boldsymbol{\sigma}_{k,i}^2\right]=\beta E\left[\boldsymbol{\sigma}_{k,i-1}^2\right]+(1-\beta)\,\sigma_k^2$$

$$
\begin{aligned}
E\left[\boldsymbol{\sigma}_{k,i}^{4}\right] &= \beta^{2} E\left[\boldsymbol{\sigma}_{k,i-1}^{4}\right] + 2\beta\left(1-\beta\right)\sigma_{k}^{2} E\left[\boldsymbol{\sigma}_{k,i-1}^{2}\right] + \left(1-\beta\right)^{2}\sigma_{k}^{4} \\
E\left[\boldsymbol{\sigma}_{k,i}^{6}\right] &= \beta^{3} E\left[\boldsymbol{\sigma}_{k,i-1}^{6}\right] + 3\beta^{2}\left(1-\beta\right)\sigma_{k}^{2} E\left[\boldsymbol{\sigma}_{k,i-1}^{4}\right] + 3\beta\left(1-\beta\right)^{2}\sigma_{k}^{4} E\left[\boldsymbol{\sigma}_{k,i-1}^{2}\right] \\
&\quad + \left(1-\beta\right)^{3}\sigma_{k}^{6} \\
E\left[\boldsymbol{\sigma}_{k,i}^{8}\right] &= \beta^{4} E\left[\boldsymbol{\sigma}_{k,i-1}^{8}\right] + 4\beta^{3}\left(1-\beta\right)\sigma_{k}^{2} E\left[\boldsymbol{\sigma}_{k,i-1}^{6}\right] + 6\beta^{2}\left(1-\beta\right)^{2}\sigma_{k}^{4} E\left[\boldsymbol{\sigma}_{k,i-1}^{4}\right] \\
&\quad + 4\beta\left(1-\beta\right)^{3}\sigma_{k}^{6} E\left[\boldsymbol{\sigma}_{k,i-1}^{2}\right] + \left(1-\beta\right)^{4}\sigma_{k}^{8} \qquad \text{(A.33)}
\end{aligned}
$$

Global MSD:

$$
\eta(i) = \eta(i-1) + b^{T}\bar{F}^{i} q_{\eta} - \parallel \bar{w}^{(o)} \parallel^{2}_{\bar{F}^{i}\left(I-\bar{F}\right)q_{\eta}} \qquad \text{(A.34)}
$$

A.6 New Variable Step-Size Transform-Domain DLMS (NVSTDLMS)

The variable step-size based on NVSTDLMS procedure for single-task networks is summarized in Algorithm A.6 (A) for easy simulation purposes.

Algorithm A.6 (A): ATC Diffusion LMS for New Variable Step-Size Transform-Domain (NVSTDLMS) Single-Task Networks Algorithm

---

Initialize: $\phi_{k,0} = 0$ for all $k = 1, ..., N$ and $u_{k,i}$ in transform domain.

Procedure: Start "for loop" at iteration $i \geq 1$ and node $1 \leq k \leq N$

Adapt Step:

$$\psi_{k,i+1} = \phi_{k,i} + \frac{\mu_k(i)}{\delta + \sigma^2_{k,i}} u^*_{k,i}(\, d_k(i) - u_{k,i}\phi_{k,i}) \tag{A.35}$$

Update Power Estimate:

$$\sigma^2_{k,i+1} = \beta\sigma^2_{k,i} + (\, 1 - \beta)\, |u_{k,i}|^2 \tag{A.36}$$

Update Step-Size:

$$\mu_k(i+1) = \begin{cases} \frac{\varepsilon_{1,k}(i)}{\varepsilon_{2,k}(i)}, & \text{if } \mu_k(i+1) \in (\, \mu_{k,min}, \mu_{k,max}) \\ \mu_{k,max}, & \text{if } \mu_k(i+1) \geq \mu_{k,max} \\ \mu_{k,min}, & \text{if } \mu_k(i+1) \leq \mu_{k,min} \end{cases} \tag{A.37}$$

Combine Step:

$$\phi_{k,i+1} = \sum_{l \in \mathcal{N}_k} c_{kl}\psi_{l,i+1} \tag{A.38}$$

The theoretical derivation of variable step-size based on NVSTDLMS that considers the aforementioned equations for mean transient and mean-square analyses in order to formulate single-task networks is short listed in Algorithm

A.6 (B) for easy reference.

Algorithm A.6 (B): Theoretical MSD of ATC Diffusion LMS for New Variable Step-Size Transform-Domain (NVSTDLMS) Single-Task Networks.

Initialize: $\eta(0) = \| \bar{w}^{(o)} \|^2$ where $\bar{w}^{(o)}$ in transform domain and $\bar{F}^0 = I_{N^2M^2}$.

$$\bar{F} = \left(\bar{G}^* \odot \bar{G}^{*T}\right) [\, I_{N^2M^2} - (\, I_{NM} \odot \Lambda E\left[\boldsymbol{D}_{i-1}\right]) \\ - (\, \Lambda E\left[\boldsymbol{D}_{i-1}\right] \odot I_{NM}) \, + E\left[\boldsymbol{D}_{i-1} \odot \boldsymbol{D}_{i-1}\right] \mathcal{A}] \quad (N^2M^2 \times N^2M^2) \tag{A.39}$$

Procedure: Start "for loop" at iteration $i \geq 1$ and node $1 \leq k \leq N$

Update Step-Size:

$$E\left[\boldsymbol{D}_i^2\right] = diag\{E\left[\boldsymbol{\mu}_1^{'}(i)\right] I_M, E\left[\boldsymbol{\mu}_2^{'}(i)\right] I_M, ..., E\left[\boldsymbol{\mu}_N^{'}(i)\right] I_M\} \\ E\left[\boldsymbol{D}_i\right] = diag\{E\left[\boldsymbol{\mu}_1^{'2}(i)\right] I_M, E\left[\boldsymbol{\mu}_2^{'2}(i)\right] I_M, ..., E\left[\boldsymbol{\mu}_N^{'2}(i)\right] I_M\} \tag{A.40}$$

$$E\left[\boldsymbol{\mu}_k^{'}(i)\right] = E\left[\boldsymbol{\mu}_k(i)\right] \left[\frac{1}{E\left[\boldsymbol{\sigma}_{k,i}^2\right]} - \frac{\delta}{E\left[\boldsymbol{\sigma}_{k,i}^4\right]}\right] \\ E\left[\boldsymbol{\mu}_k^{'2}(i)\right] = E\left[\boldsymbol{\mu}_k^2(i)\right] \left[\frac{1}{E\left[\boldsymbol{\sigma}_{k,i}^4\right]} + \frac{\delta^2}{E\left[\boldsymbol{\sigma}_{k,i}^8\right]} - \frac{2\delta}{E\left[\boldsymbol{\sigma}_{k,i}^6\right]}\right] \tag{A.41}$$

$$
\begin{aligned}
E\left[\boldsymbol{\mu}_k(i)\right] &= \frac{\alpha_1 E\left[\boldsymbol{\varepsilon}_{1,k}(i-1)\right] + E\left[\boldsymbol{e}_k^2(i)\right]}{\alpha_2 E\left[\boldsymbol{\varepsilon}_{2,k}(i-1)\right] + E\left[\boldsymbol{e}_k^2(i)\right]} \\
E\left[\boldsymbol{\mu}_k^2(i)\right] &= \frac{\alpha_1^2 E\left[\boldsymbol{\varepsilon}_{1,k}^2(i-1)\right] + 2\alpha_1 E\left[\boldsymbol{\varepsilon}_{1,k}(i-1)\right] E\left[\boldsymbol{e}_k^2(i)\right] + E\left[\boldsymbol{e}_k^4(i)\right]}{\alpha_2^2 E\left[\boldsymbol{\varepsilon}_{2,k}^2(i-1)\right] + 2\alpha_2 E\left[\boldsymbol{\varepsilon}_{2,k}(i-1)\right] E\left[\boldsymbol{e}_k^2(i)\right] + E\left[\boldsymbol{e}_k^4(i)\right]}
\end{aligned}
\tag{A.42}
$$

$$
\begin{aligned}
E\left[\boldsymbol{\sigma}_{k,i}^2\right] &= \beta E\left[\boldsymbol{\sigma}_{k,i-1}^2\right] + (1-\beta)\,\sigma_k^2 \\
E\left[\boldsymbol{\sigma}_{k,i}^4\right] &= \beta^2 E\left[\boldsymbol{\sigma}_{k,i-1}^4\right] + 2\beta\,(1-\beta)\,\sigma_k^2 E\left[\boldsymbol{\sigma}_{k,i-1}^2\right] + (1-\beta)^2\,\sigma_k^4 \\
E\left[\boldsymbol{\sigma}_{k,i}^6\right] &= \beta^3 E\left[\boldsymbol{\sigma}_{k,i-1}^6\right] + 3\beta^2\,(1-\beta)\,\sigma_k^2 E\left[\boldsymbol{\sigma}_{k,i-1}^4\right] + 3\beta\,(1-\beta)^2\,\sigma_k^4 E\left[\boldsymbol{\sigma}_{k,i-1}^2\right] \\
&\quad + (1-\beta)^3\,\sigma_k^6 \\
E\left[\boldsymbol{\sigma}_{k,i}^8\right] &= \beta^4 E\left[\boldsymbol{\sigma}_{k,i-1}^8\right] + 4\beta^3\,(1-\beta)\,\sigma_k^2 E\left[\boldsymbol{\sigma}_{k,i-1}^6\right] + 6\beta^2\,(1-\beta)^2\,\sigma_k^4 E\left[\boldsymbol{\sigma}_{k,i-1}^4\right] \\
&\quad + 4\beta\,(1-\beta)^3\,\sigma_k^6 E\left[\boldsymbol{\sigma}_{k,i-1}^2\right] + (1-\beta)^4\,\sigma_k^8
\end{aligned}
\tag{A.43}
$$

Global MSD:

$$
\eta(i) = \eta(i-1) + b^T \bar{F}^i q_\eta - \parallel \bar{w}^{(o)} \parallel^2_{\bar{F}^i\left(I-\bar{F}\right)q_\eta} \tag{A.44}
$$

A.7 Optimal Variable Step-Size Transform-Domain DLMS (VSSTDLMS)

The optimal variable step-size transform domain DLMS (VSSTDLMS) procedure for single-task networks is summarized in Algorithm A.7 (A) for easy simulation purposes.

Algorithm A.7 (A): ATC Diffusion LMS for Optimal Variable Step-Size Transform Domain (VSSTDLMS) Single-Task Networks Algorithm

---

Initialize: $\phi_{k,0} = 0$ for all $k = 1, ..., N$ and $u_{k,i}$ in transform domain.

Procedure: Start "for loop" at iteration $i \geq 1$ and node $1 \leq k \leq N$

Adapt Step:

$$\psi_{k,i+1} = \phi_{k,i} + \frac{\mu_k(i)}{\delta + \sigma_{k,i}^2} u_{k,i}^* (d_k(i) - u_{k,i}\phi_{k,i}) \tag{A.45}$$

Update Power Estimate:

$$\sigma_{k,i+1}^2 = \beta \sigma_{k,i}^2 + (1 - \beta) |u_{k,i}|^2 \tag{A.46}$$

Update Step-Size:

$$\rho_k(i+1) = \gamma \rho_k(i) + (1 - \gamma) \left(e_k^2(i) \hat{u}_{k,i}^T \hat{u}_{k,i} \right)$$

$$\eta_k(i+1) = \gamma \eta_k(i) + (1-\gamma)\ \big(e_k(i) e_k(i-1)\big)$$
$$\mu_k(i+1) = \frac{\big|\eta_k(i+1)\big|}{\rho_k(i+1)} \tag{A.47}$$

Combine Step:

$$\phi_{k,i+1} = \sum_{l \in \mathcal{N}_k} c_{kl} \psi_{l,i+1} \tag{A.48}$$

The theoretical derivation of optimal variable step-size transform domain (VSST-DLMS) that considers the aforementioned equations for mean transient and mean-square analyses in order to formulate single-task networks is short listed in Algorithm A.7 (B) for easy reference.

Algorithm A.7 (B): Theoretical MSD of ATC Diffusion LMS for Optimal Variable Step-Size Transform Domain (VSSTDLMS) Single-Task Networks.

Initialize: $\eta(0) = \parallel \bar{w}^{(o)} \parallel^2$ where $\bar{w}^{(o)}$ in transform domain and $\bar{F}^0 = I_{N^2M^2}$.

$$\bar{F} = \left(\bar{G}^* \odot \bar{G}^{*T}\right) [\, I_{N^2M^2} - (\, I_{NM} \odot \Lambda E\, [\boldsymbol{D}_{i-1}])$$
$$- (\, \Lambda E\, [\boldsymbol{D}_{i-1}] \odot I_{NM}) \, + E\, [\boldsymbol{D}_{i-1} \odot \boldsymbol{D}_{i-1}]\, \mathcal{A}] \quad (N^2M^2 \times N^2M^2) \tag{A.49}$$

Procedure: Start "for loop" at iteration $i \geq 1$ and node $1 \leq k \leq N$

Update Step-Size:

$$E\left[\boldsymbol{D}_i^2\right] = diag\{E\left[\boldsymbol{\mu}_1^{'}(i)\right] I_M, E\left[\boldsymbol{\mu}_2^{'}(i)\right] I_M, ..., E\left[\boldsymbol{\mu}_N^{'}(i)\right] I_M\}$$
$$E\left[\boldsymbol{D}_i\right] = diag\{E\left[\boldsymbol{\mu}_1^{'2}(i)\right] I_M, E\left[\boldsymbol{\mu}_2^{'2}(i)\right] I_M, ..., E\left[\boldsymbol{\mu}_N^{'2}(i)\right] I_M\} \tag{A.50}$$

$$E\left[\boldsymbol{\mu}_k^{'}(i)\right] = E\left[\boldsymbol{\mu}_k(i)\right]\left[\frac{1}{E\left[\boldsymbol{\sigma}_{k,i}^2\right]} - \frac{\delta}{E\left[\boldsymbol{\sigma}_{k,i}^4\right]}\right]$$
$$E\left[\boldsymbol{\mu}_k^{'2}(i)\right] = E\left[\boldsymbol{\mu}_k^2(i)\right]\left[\frac{1}{E\left[\boldsymbol{\sigma}_{k,i}^4\right]} + \frac{\delta^2}{E\left[\boldsymbol{\sigma}_{k,i}^8\right]} - \frac{2\delta}{E\left[\boldsymbol{\sigma}_{k,i}^6\right]}\right] \tag{A.51}$$

$$\begin{aligned}
E\left[\boldsymbol{\sigma}_{k,i}^2\right] &= \beta E\left[\boldsymbol{\sigma}_{k,i-1}^2\right] + (1-\beta)\,\sigma_k^2 \\
E\left[\boldsymbol{\sigma}_{k,i}^4\right] &= \beta^2 E\left[\boldsymbol{\sigma}_{k,i-1}^4\right] + 2\beta(1-\beta)\,\sigma_k^2 E\left[\boldsymbol{\sigma}_{k,i-1}^2\right] + (1-\beta)^2\sigma_k^4 \\
E\left[\boldsymbol{\sigma}_{k,i}^6\right] &= \beta^3 E\left[\boldsymbol{\sigma}_{k,i-1}^6\right] + 3\beta^2(1-\beta)\,\sigma_k^2 E\left[\boldsymbol{\sigma}_{k,i-1}^4\right] + 3\beta(1-\beta)^2\sigma_k^4 E\left[\boldsymbol{\sigma}_{k,i-1}^2\right] \\
&\quad + (1-\beta)^3\sigma_k^6 \\
E\left[\boldsymbol{\sigma}_{k,i}^8\right] &= \beta^4 E\left[\boldsymbol{\sigma}_{k,i-1}^8\right] + 4\beta^3(1-\beta)\,\sigma_k^2 E\left[\boldsymbol{\sigma}_{k,i-1}^6\right] + 6\beta^2(1-\beta)^2\sigma_k^4 E\left[\boldsymbol{\sigma}_{k,i-1}^4\right] \\
&\quad + 4\beta(1-\beta)^3\sigma_k^6 E\left[\boldsymbol{\sigma}_{k,i-1}^2\right] + (1-\beta)^4\sigma_k^8
\end{aligned} \tag{A.52}$$

$$\eta(i) = \eta(i-1) + b^T\bar{F}^i q_\eta - \parallel \bar{w}^{(o)} \parallel^2_{\bar{F}^i\left(I-\bar{F}\right)q_\eta} \tag{A.53}$$

Appendix-B Multi-Task Networks Algorithms

The list of theoretical and simulation procedures of the following algorithms are summarized in appendix-B. The algorithms are

1. ATC DLMS with adaptive combiners
2. VSSLMS with adaptive combiners
3. TDLMS with adaptive combiners
4. DCTLMS with adaptive combiners
5. TDVSS with adaptive combiners
6. NVSTDLMS with adaptive combiners
7. VSSTDLMS with adaptive combiners

It is a useful practice when any multi-task networks algorithm is simulated.

B.1 Adapt-Then-Combine DLMS with Adaptive Combiners

The Multi-Task DLMS (DLMS) procedure with Adaptive Combiners technique is summarized in Algorithm B.1 (A) for easy simulation purposes.

Algorithm B.1 (A): ATC Diffusion LMS Multi-Task Networks with Adaptive Combiner

---

Initialize: $C(0) = I_N$ and $\phi_{k,0} = 0$ for all $k = 1, ..., N$.

Procedure: Start "for loop" at iteration $i \geq 1$ and node $1 \leq k \leq N$

Adapt Step:

$$\psi_{k,i+1} = \phi_{k,i} + \mu_k u_{k,i}^* (d_k(i) - u_{k,i}\phi_{k,i}) \tag{B.1}$$

Update Clustering:

$$\begin{aligned} q_{k,i} &= u_{k,i}^* (d_k(i) - u_{k,i}\psi_{k,i+1}) \\ c_{kl}(i+1) &= \frac{\| \psi_{k,i+1} + \mu_k q_{k,i} - \psi_{l,i+1} \|^{-2}}{\sum_{j \in \mathcal{N}_k} \| \psi_{k,i+1} + \mu_k q_{k,i} - \psi_{j,i+1} \|^{-2}} \end{aligned} \tag{B.2}$$

Combine Step:

$$\phi_{k,i+1} = \sum_{l \in \mathcal{N}_k} c_{kl}(i+1)\psi_{l,i+1} \tag{B.3}$$

The theoretical derivation of Adaptive Combiner that considers the aforementioned equations for mean transient and mean-square analyses in order to achieve multi-task adaptive filter is short listed in Algorithm B.1 (B) for easy reference.

Algorithm B.1 (B): Theoretical MSD of ATC Diffusion LMS (DLMS) Multi-Task Networks with Adaptive Combiner

Initialize: $C(0) = I_N$, $\eta(0) =\| \bar{w}^{(o)} \|^2$, $\bar{F}^0 = I_{N^2M^2}$, $\bar{F}_1^0 = I_{NM}$, $\eta_k(0) =\| \bar{w}^{(o)} \|^2$ and $\zeta_k(0) = \bar{w}^o$ for all $k = 1, ..., N$.

$$\begin{aligned}\bar{F} = E\left[\bar{\boldsymbol{G}}^*_{i-1} \odot \bar{\boldsymbol{G}}^{*T}_{i-1}\right] [\, I_{N^2M^2} - (\, I_{NM} \odot \Lambda D) \\ - (\, \Lambda D \odot I_{NM}) + (\, D \odot D)\mathcal{A}] \quad (N^2M^2 \times N^2M^2)\end{aligned} \tag{B.4}$$

$$\bar{F}_1 = [\, I_{NM} - D\Lambda]\, E\, [\boldsymbol{C}_{i-1}] \quad (NM \times NM) \tag{B.5}$$

Procedure: Start "for loop" at iteration $i \geq 1$ and node $1 \leq k \leq N$

Local node MSD:

$$\bar{\sigma}_k = \text{bvec}\{\mathbf{0}_{(k-1)M}, I_M, \mathbf{0}_{(N-k)M}\} \triangleq q_{k\eta} \quad (N^2M^2 \times 1) \tag{B.6}$$

$$\eta_k(i) = \eta_k(i-1) + b^T \bar{F}^i q_{k\eta} - \parallel \bar{w}^{(o)} \parallel^2_{\bar{F}^i \left(I - \bar{F}\right) q_{k\eta}}$$

$$\eta_k^{'}(i) = \eta_k(i) + b^T \bar{F}^i q_{k\eta} - \parallel \bar{w}^{(o)} \parallel^2_{\bar{F}^i \left(I - \bar{F}\right) q_{k\eta}} \tag{B.7}$$

Local node mean:

$$q_{1,k\eta} = \text{bvec}\{\mathbf{0}_{(k-1)M}, 1_M, \mathbf{0}_{(N-k)M}\} \quad (NM \times 1) \tag{B.8}$$

$$\zeta_k(i) = \zeta_k(i-1) - q^T_{1,k\eta} \left(I_{NM} - \bar{F}_1\right) \bar{F}_1^i w^o$$

$$\zeta_k^{'}(i) = \zeta_k(i) - q^T_{1,k\eta} \left(I_{NM} - \bar{F}_1\right) \bar{F}_1^i w^o \tag{B.9}$$

Update Step:

$$c_{kl}(i+1) = \frac{\left(\eta_k^{'}(i+1) + \eta_l(i+1) - 2\zeta_k^{'}(i+1)\zeta_l(i+1)\right)^{-1}}{\sum_{j \in \mathcal{N}_k} \left(\eta_k^{'}(i) + \eta_j(i+1) - 2\zeta_k^{'}(i+1)\zeta_j(i+1)\right)^{-1}} \tag{B.10}$$

Global MSD:

$$\eta(i) = \eta(i-1) + b^T \bar{F}^i q_\eta - \parallel \bar{w}^{(o)} \parallel^2_{\bar{F}^i \left(I - \bar{F}\right) q_\eta} \tag{B.11}$$

B.2 Variable Step-Size DLMS (VSSLMS) with Adaptive Combiners

The algorithm of variable step-size DLMS (VSSLMS) procedure for Multi-Task DLMS with adaptive combiner technique is summarized in Algorithm B.2 (A) for easy simulation purposes.

Algorithm B.2 (A): ATC Variable Step-Size DLMS Multi-Task Networks with Adaptive Combiner

---

Initialize: $C(0) = I_N$ and $\phi_{k,0} = 0$ for all $k = 1, ..., N$.

Procedure: Start "for loop" at iteration $i \geq 1$ and node $1 \leq k \leq N$

Adapt Step:

$$\psi_{k,i+1} = \phi_{k,i} + \mu_k(i)u^*_{k,i}(\, d_k(i) - u_{k,i}\phi_{k,i}) \tag{B.12}$$

Update Clustering:

$$\begin{aligned} q_{k,i} &= u^*_{k,i}(\, d_k(i) - u_{k,i}\psi_{k,i+1}) \\ c_{kl}(i+1) &= \frac{\parallel \psi_{k,i+1} + \mu_k(i)q_{k,i} - \psi_{l,i+1} \parallel^{-2}}{\sum_{j\in\mathcal{N}_k} \parallel \psi_{k,i+1} + \mu_k(i)q_{k,i} - \psi_{j,i+1} \parallel^{-2}} \end{aligned} \tag{B.13}$$

Update Step-Size:

$$\mu_k(i+1) = \alpha\mu_k(i) + \gamma e_k^2(i) \tag{B.14}$$

Combine Step:

$$\phi_{k,i+1} = \sum_{l\in\mathcal{N}_k} c_{kl}(i+1)\psi_{l,i+1} \tag{B.15}$$

The theoretical derivation of variable step-size DLMS (VSSLMS) with adaptive clustering that considers the aforementioned equations for mean transient and mean-square analyses in order to achieve multi-task adaptive filter is short listed in Algorithm B.2 (B) for easy reference.

Algorithm B.2 (B): Theoretical MSD of ATC Diffusion LMS with Adaptive Clustering for Variable Step-Size Multi-Task Networks

---

Initialize: $C(0) = I_N$, $\eta(0) = \| \bar{w}^{(o)} \|^2$, $\bar{F}^0 = I_{N^2M^2}$, $\bar{F}_1^0 = I_{NM}$, $\eta_k(0) = \| \bar{w}^{(o)} \|^2$ and $\zeta_k(0) = \bar{w}^o$ for all $k = 1, ..., N$.

$$\begin{aligned}\bar{F} = E\left[\bar{\boldsymbol{G}}_{i-1}^{*} \odot \bar{\boldsymbol{G}}_{i-1}^{*T}\right]\left[\, I_{N^2M^2} - (\, I_{NM} \odot \Lambda E\left[\boldsymbol{D}_{i-1}\right])\right.\\ \left. - (\,\Lambda E\left[\boldsymbol{D}_{i-1}\right] \odot I_{NM}) + E\left[\boldsymbol{D}_{i-1} \odot \boldsymbol{D}_{i-1}\right]\mathcal{A}\right] \quad (N^2M^2 \times N^2M^2)\end{aligned} \tag{B.16}$$

$$\bar{F}_1 = [\, I_{NM} - E\,[\boldsymbol{D}_{i-1}]\,\Lambda]\, E\,[\boldsymbol{C}_{i-1}] \quad (NM \times NM) \tag{B.17}$$

Procedure: Start "for loop" at iteration $i \geq 1$ and node $1 \leq k \leq N$

Local node MSD:

$$\bar{\sigma}_k = \text{bvec}\{\mathbf{0}_{(k-1)M}, I_M, \mathbf{0}_{(N-k)M}\} \triangleq q_{k\eta} \quad (N^2M^2 \times 1) \tag{B.18}$$

$$\begin{aligned} \eta_k(i) &= \eta_k(i-1) + b^T \bar{F}^i q_{k\eta} - \parallel \bar{w}^{(o)} \parallel^2_{\bar{F}^i \left(I - \bar{F}\right) q_{k\eta}} \\ \eta_k^{'}(i) &= \eta_k(i) + b^T \bar{F}^i q_{k\eta} - \parallel \bar{w}^{(o)} \parallel^2_{\bar{F}^i \left(I - \bar{F}\right) q_{k\eta}} \end{aligned} \tag{B.19}$$

Local node mean:

$$q_{1,k\eta} = \text{bvec}\{\mathbf{0}_{(k-1)M}, 1_M, \mathbf{0}_{(N-k)M}\} \quad (NM \times 1) \tag{B.20}$$

$$\begin{aligned} \zeta_k(i) &= \zeta_k(i-1) - q_{1,k\eta} \left(I_N - \bar{F}_1\right) \bar{F}_1^i w^o \\ \zeta_k^{'}(i) &= \zeta_k(i) - q_{1,k\eta} \left(I_N - \bar{F}_1\right) \bar{F}_1^i w^o \end{aligned} \tag{B.21}$$

Update Clustering:

$$c_{kl}(i+1) = \frac{\left(\eta_k^{'}(i+1) + \eta_l(i+1) - 2\zeta_k^{'}(i+1)\zeta_l(i+1)\right)^{-1}}{\sum_{j \in \mathcal{N}_k} \left(\eta_k^{'}(i) + \eta_j(i+1) - 2\zeta_k^{'}(i+1)\zeta_j(i+1)\right)^{-1}} \tag{B.22}$$

Update Step-Size:

$$E\left[\boldsymbol{D}_i^2\right] = \alpha^2 E\left[\boldsymbol{D}_{i-1}^2\right] + 2\alpha\gamma E\left[\boldsymbol{D}_{i-1}\right] E\left[\boldsymbol{E}^{i-1}\right] + \gamma^2 E\left[\boldsymbol{E}^{2,i-1}\right]$$
$$E\left[\boldsymbol{D}_i\right] = \alpha E\left[\boldsymbol{D}_{i-1}\right] + \gamma E\left[\boldsymbol{E}^{i-1}\right] \tag{B.23}$$

Global MSD:

$$\eta(i) = \eta(i-1) + b^T \bar{F}^i q_\eta - \parallel \bar{w}^{(o)} \parallel^2_{\bar{F}^i\left(I-\bar{F}\right)q_\eta} \tag{B.24}$$

B.3 Transform Domain DLMS (TDLMS) with Adaptive Combiners

The transform domain DLMS (TDLMS) procedure for Multi-Task Networks is summarized in Algorithm B.3 (A) for easy simulation purposes.

Algorithm B.3 (A): ATC Diffusion LMS for Transform-Domain DLMS (TDLMS) Multi-Task Networks Algorithm

---

Initialize: $C(0) = I_N$, $\phi_{k,0} = 0$ for all $k = 1, ..., N$, and $u_{k,i}$ in transform domain.

Procedure: Start "for loop" at iteration $i \geq 1$ and node $1 \leq k \leq N$

Adapt Step:

$$\psi_{k,i+1} = \phi_{k,i} + \frac{\mu_k}{\delta + \sigma^2_{k,i}} u^*_{k,i} (\, d_k(i) - u_{k,i}\phi_{k,i}) \tag{B.25}$$

Update Power Estimate:

$$\sigma^2_{k,i+1} = \beta\sigma^2_{k,i} + (\, 1 - \beta)\, |u_{k,i}|^2 \tag{B.26}$$

Update Clustering:

$$\begin{aligned} q_{k,i} &= u^*_{k,i} (\, d_k(i) - u_{k,i}\psi_{k,i+1}) \\ c_{kl}(i+1) &= \frac{\|\, \psi_{k,i+1} + \mu_k q_{k,i} - \psi_{l,i+1}\, \|^{-2}}{\sum_{j \in \mathcal{N}_k} \|\, \psi_{k,i+1} + \mu_k q_{k,i} - \psi_{j,i+1}\, \|^{-2}} \end{aligned} \tag{B.27}$$

Combine Step:

$$\phi_{k,i+1} = \sum_{l \in \mathcal{N}_k} c_{kl}(i+1)\psi_{l,i+1} \tag{B.28}$$

The theoretical derivation of variable step-size transform domain DLMS (TDLMS) with Adaptive Combiners that considers the aforementioned equations for mean transient and mean-square analyses in order to achieve Multi-Task Networks is short listed in Algorithm B.3 (B) for easy reference.

Algorithm B.3 (B): Theoretical MSD of ATC Diffusion LMS for Transform Domain DLMS (TDLMS) Multi-Task Networks with Adaptive Combiner

Initialize: $C(0) = I_N$, $\eta(0) =\parallel \bar{w}^{(o)} \parallel^2$, $\bar{F}^0 = I_{N^2M^2}$, $\bar{F}_1^0 = I_{NM}$, $\eta_k(0) =\parallel \bar{w}^{(o)} \parallel^2$, $\zeta_k(0) = \bar{w}^o$ for all $k = 1, ..., N$, and $\bar{w}^o$ in transform domain.

$$\begin{aligned}\bar{F} = E\left[\bar{\boldsymbol{G}}_{i-1}^{*} \odot \bar{\boldsymbol{G}}_{i-1}^{*T}\right] [\, I_{N^2M^2} - (\, I_{NM} \odot \Lambda E\,[\boldsymbol{D}_{i-1}]) \\ - (\,\Lambda E\,[\boldsymbol{D}_{i-1}] \odot I_{NM}) + E\,[\boldsymbol{D}_{i-1} \odot \boldsymbol{D}_{i-1}]\,\mathcal{A}] \quad (N^2M^2 \times N^2M^2)\end{aligned} \tag{B.29}$$

$$\bar{F}_1 = [\, I_{NM} - E\,[\boldsymbol{D}_{i-1}]\,\Lambda]\, E\,[\boldsymbol{C}_{i-1}] \quad (NM \times NM) \tag{B.30}$$

Procedure: Start "for loop" at iteration $i \geq 1$ and node $1 \leq k \leq N$

Local node MSD:

$$\bar{\sigma}_k = \text{bvec}\{\mathbf{0}_{(k-1)M}, I_M, \mathbf{0}_{(N-k)M}\} \triangleq q_{k\eta} \quad (N^2M^2 \times 1) \tag{B.31}$$

$$\begin{aligned}\eta_k(i) &= \eta_k(i-1) + b^T \bar{F}^i q_{k\eta} - \parallel \bar{w}^{(o)} \parallel^2_{\bar{F}^i\left(I-\bar{F}\right)q_{k\eta}} \\ \eta_k^{'}(i) &= \eta_k(i) + b^T \bar{F}^i q_{k\eta} - \parallel \bar{w}^{(o)} \parallel^2_{\bar{F}^i\left(I-\bar{F}\right)q_{k\eta}}\end{aligned} \tag{B.32}$$

Local node mean:

$$q_{1,k\eta} = \text{bvec}\{\mathbf{0}_{(k-1)M}, 1_M, \mathbf{0}_{(N-k)M}\} \quad (NM \times 1) \tag{B.33}$$

$$\begin{aligned}\zeta_k(i) &= \zeta_k(i-1) - q_{1,k\eta}\left(I_N - \bar{F}_1\right)\bar{F}_1^i w^o \\ \zeta_k^{'}(i) &= \zeta_k(i) - q_{1,k\eta}\left(I_N - \bar{F}_1\right)\bar{F}_1^i w^o\end{aligned} \tag{B.34}$$

Update Clustering:

$$c_{kl}(i+1) = \frac{\left(\eta_k^{'}(i+1) + \eta_l(i+1) - 2\zeta_k^{'}(i+1)\zeta_l(i+1)\right)^{-1}}{\sum_{j\in\mathcal{N}_k}\left(\eta_k^{'}(i) + \eta_j(i+1) - 2\zeta_k^{'}(i+1)\zeta_j(i+1)\right)^{-1}} \tag{B.35}$$

Update Step-Size:

$$\begin{aligned}E\left[\boldsymbol{D}_i^2\right] &= diag\{E\left[\boldsymbol{\mu}_1^{'}(i)\right]I_M, E\left[\boldsymbol{\mu}_2^{'}(i)\right]I_M, ..., E\left[\boldsymbol{\mu}_N^{'}(i)\right]I_M\} \\ E\left[\boldsymbol{D}_i\right] &= diag\{E\left[\boldsymbol{\mu}_1^{'2}(i)\right]I_M, E\left[\boldsymbol{\mu}_2^{'2}(i)\right]I_M, ..., E\left[\boldsymbol{\mu}_N^{'2}(i)\right]I_M\}\end{aligned} \tag{B.36}$$

$$\begin{aligned}E\left[\boldsymbol{\mu}_k^{'}(i)\right] &= \mu_k\left[\frac{1}{E\left[\boldsymbol{\sigma}_{k,i}^2\right]} - \frac{\delta}{E\left[\boldsymbol{\sigma}_{k,i}^4\right]}\right] \\ E\left[\boldsymbol{\mu}_k^{'2}(i)\right] &= \mu_k^2\left[\frac{1}{E\left[\boldsymbol{\sigma}_{k,i}^4\right]} + \frac{\delta^2}{E\left[\boldsymbol{\sigma}_{k,i}^8\right]} - \frac{2\delta}{E\left[\boldsymbol{\sigma}_{k,i}^6\right]}\right]\end{aligned} \tag{B.37}$$

$$E\left[\boldsymbol{\sigma}_{k,i}^2\right] = \beta E\left[\boldsymbol{\sigma}_{k,i-1}^2\right] + (1-\beta)\,\sigma_k^2$$

$$E\left[\boldsymbol{\sigma}_{k,i}^4\right] = \beta^2 E\left[\boldsymbol{\sigma}_{k,i-1}^4\right] + 2\beta(1-\beta)\,\sigma_k^2 E\left[\boldsymbol{\sigma}_{k,i-1}^2\right] + (1-\beta)^2\,\sigma_k^4$$

$$E\left[\boldsymbol{\sigma}_{k,i}^6\right] = \beta^3 E\left[\boldsymbol{\sigma}_{k,i-1}^6\right] + 3\beta^2(1-\beta)\,\sigma_k^2 E\left[\boldsymbol{\sigma}_{k,i-1}^4\right] + 3\beta(1-\beta)^2\,\sigma_k^4 E\left[\boldsymbol{\sigma}_{k,i-1}^2\right] + (1-\beta)^3\,\sigma_k^6$$

$$E\left[\boldsymbol{\sigma}_{k,i}^8\right] = \beta^4 E\left[\boldsymbol{\sigma}_{k,i-1}^8\right] + 4\beta^3(1-\beta)\,\sigma_k^2 E\left[\boldsymbol{\sigma}_{k,i-1}^6\right] + 6\beta^2(1-\beta)^2\,\sigma_k^4 E\left[\boldsymbol{\sigma}_{k,i-1}^4\right] + 4\beta(1-\beta)^3\,\sigma_k^6 E\left[\boldsymbol{\sigma}_{k,i-1}^2\right] + (1-\beta)^4\,\sigma_k^8 \tag{B.38}$$

Global MSD:

$$\eta(i) = \eta(i-1) + b^T \bar{F}^i q_\eta - \parallel \bar{w}^{(o)} \parallel^2_{\bar{F}^i\left(I-\bar{F}\right)q_\eta} \tag{B.39}$$

B.4 Discrete Cosine Transform DLMS (DCTLMS) with Adaptive Combiners

The variable step-size based on DCTLMS procedure for Multi-Task Networks with adaptive clustering technique is summarized in Algorithm B.4 (A) for easy simulation purposes.

Algorithm B.4 (A): ATC Diffusion LMS with Adaptive Clustering for Variable Step-Size Multi-Task Networks Based on Discrete Cosine Transform DLMS (DCTLMS)

Initialize: $C(0) = I_N$, $\phi_{k,0} = 0$ for all $k = 1, ..., N$, and $u_{k,i}$ in transform domain.

Procedure: Start "for loop" at iteration $i \geq 1$ and node $1 \leq k \leq N$

Adapt Step:

$$\psi_{k,i+1} = \phi_{k,i} + \mu_k(i)u^*_{k,i}(\, d_k(i) - u_{k,i}\phi_{k,i}) \tag{B.40}$$

Update Clustering:

$$\begin{aligned} q_{k,i} &= u^*_{k,i}(\, d_k(i) - u_{k,i}\psi_{k,i+1}) \\ c_{kl}(i+1) &= \frac{\parallel \psi_{k,i+1} + \mu_k(i)q_{k,i} - \psi_{l,i+1} \parallel^{-2}}{\sum_{j\in\mathcal{N}_k} \parallel \psi_{k,i+1} + \mu_k(i)q_{k,i} - \psi_{j,i+1} \parallel^{-2}} \end{aligned} \tag{B.41}$$

Update Step-Size:

$$\mu_k(i+1) = \beta\mu_k(i) + \gamma(\,1-\beta)\,\left(\frac{1}{\delta + \frac{1}{L}u'^*_{k,i}u'_{k,i}}\right) \tag{B.42}$$

Combine Step:

$$\phi_{k,i+1} = \sum_{l\in\mathcal{N}_k} c_{kl}(i+1)\psi_{l,i+1} \tag{B.43}$$

The theoretical derivation of variable step-size based on DCTLMS with adaptive combiners that considers the aforementioned equations for mean transient and mean-square analyses in order to achieve multi-task networks is short listed in Algorithm B.4 (B)for easy reference.

Algorithm B.4 (B): Theoretical MSD of ATC Diffusion LMS Variable Step-Size Multi-Task Networks Based on Discrete Cosine Transform DLMS (DCTLMS) with Adaptive Combiner

---

Initialize: $C(0) = I_N$, $\eta(0) = \| \bar{w}^{(o)} \|^2$, $\bar{F}^0 = I_{N^2M^2}$, $\bar{F}_1^0 = I_{NM}$, $\eta_k(0) = \| \bar{w}^{(o)} \|^2$, $\zeta_k(0) = \bar{w}^o$ for all $k = 1, ..., N$, and $\bar{w}^o$ in transform domain.

$$\bar{F} = E\left[\bar{\boldsymbol{G}}_{i-1}^{*} \odot \bar{\boldsymbol{G}}_{i-1}^{*T}\right] [\, I_{N^2M^2} - (\, I_{NM} \odot \Lambda E\,[\boldsymbol{D}_{i-1}]) \\ - (\, \Lambda E\,[\boldsymbol{D}_{i-1}] \odot I_{NM}) + E\,[\boldsymbol{D}_{i-1} \odot \boldsymbol{D}_{i-1}]\,\mathcal{A}] \quad (N^2M^2 \times N^2M^2) \tag{B.44}$$

$$\bar{F}_1 = [\, I_{NM} - E\,[\boldsymbol{D}_{i-1}]\,\Lambda]\, E\,[\boldsymbol{C}_{i-1}] \quad (NM \times NM) \tag{B.45}$$

Procedure: Start "for loop" at iteration $i \geq 1$ and node $1 \leq k \leq N$

Local node MSD:

$$\bar{\sigma}_k = \text{bvec}\{\mathbf{0}_{(k-1)M}, I_M, \mathbf{0}_{(N-k)M}\} \triangleq q_{k\eta} \quad (N^2M^2 \times 1) \tag{B.46}$$

$$
\begin{aligned}
\eta_k(i) &= \eta_k(i-1) + b^T \bar{F}^i q_{k\eta} - \parallel \bar{w}^{(o)} \parallel^2_{\bar{F}^i\left(I-\bar{F}\right)q_{k\eta}} \\
\eta_k^{'}(i) &= \eta_k(i) + b^T \bar{F}^i q_{k\eta} - \parallel \bar{w}^{(o)} \parallel^2_{\bar{F}^i\left(I-\bar{F}\right)q_{k\eta}}
\end{aligned} \tag{B.47}
$$

Local node mean:

$$
q_{1,k\eta} = \text{bvec}\{\mathbf{0}_{(k-1)M}, 1_M, \mathbf{0}_{(N-k)M}\} \quad (NM \times 1) \tag{B.48}
$$

$$
\begin{aligned}
\zeta_k(i) &= \zeta_k(i-1) - q_{1,k\eta}\left(I_N - \bar{F}_1\right)\bar{F}_1^i w^o \\
\zeta_k^{'}(i) &= \zeta_k(i) - q_{1,k\eta}\left(I_N - \bar{F}_1\right)\bar{F}_1^i w^o
\end{aligned} \tag{B.49}
$$

Update Clustering:

$$
c_{kl}(i+1) = \frac{\left(\eta_k^{'}(i+1) + \eta_l(i+1) - 2\zeta_k^{'}(i+1)\zeta_l(i+1)\right)^{-1}}{\sum_{j\in\mathcal{N}_k}\left(\eta_k^{'}(i) + \eta_j(i+1) - 2\zeta_k^{'}(i+1)\zeta_j(i+1)\right)^{-1}} \tag{B.50}
$$

Update Step-Size:

$$
\begin{aligned}
E\left[\boldsymbol{\mu}_{k,i+1}\right] &= \beta E\left[\boldsymbol{\mu}_{k,i}\right] + \gamma\left(1-\beta\right)\Big(\frac{L}{\sigma_k^2\left(L-2\right)} \\
&\quad - \delta\frac{L^2}{\sigma_k^4\left(L-4\right)\left(L-2\right)}\Big) \quad (M \times 1) \\
E\left[\boldsymbol{\mu}^2_{k,i+1}\right] &= \beta^2 E\left[\boldsymbol{\mu}^2_{k,i}\right] + 2\beta\gamma\left(1-\beta\right)E\left[\boldsymbol{\mu}_{k,i}\right]\Big(\frac{L}{\sigma_k^2\left(L-2\right)} \\
&\quad - \delta\frac{L^2}{\sigma_k^4\left(L-4\right)\left(L-2\right)}\Big) + \gamma^2\left(1-\beta\right)^2\Big(\frac{L^2}{\sigma_k^4\left(L-4\right)\left(L-2\right)} \\
&\quad - 2\delta\frac{L^3}{\sigma_k^6\left(L-6\right)\left(L-4\right)\left(L-2\right)}
\end{aligned}
$$

$$+\delta^2 \frac{L^4}{\sigma_k^8 (L-8)(L-6)(L-4)(L-2)}\Big) \quad (M \times 1) \tag{B.51}$$

Global MSD:

$$\eta(i) = \eta(i-1) + b^T \bar{F}^i q_\eta - \parallel \bar{w}^{(o)} \parallel^2_{\bar{F}^i \left(I - \bar{F}\right) q_\eta} \tag{B.52}$$

B.5 Transform Domain Variable Step-Size DLMS (TDVSS) with Adaptive Combiners

The transform domain variable step-size DLMS (TDVSS) procedure for Multi-Task Networks with adaptive combiner is summarized in Algorithm B.5 (A) for easy simulation purposes.

Algorithm B.5 (A): ATC Diffusion LMS for Transform Domain Variable Step-Size DLMS (TDVSS) Multi-Task Networks Algorithm with Adaptive Combiner

---

Initialize: $C(0) = I_N$, $\phi_{k,0} = 0$ for all $k = 1, ..., N$, and $u_{k,i}$ in transform domain.

Procedure: Start "for loop" at iteration $i \geq 1$ and node $1 \leq k \leq N$

Adapt Step:

$$\psi_{k,i+1} = \phi_{k,i} + \frac{\mu_k(i)}{\delta + \sigma_{k,i}^2} u_{k,i}^* (\, d_k(i) - u_{k,i}\phi_{k,i}) \tag{B.53}$$

Update Power Estimate:

$$\sigma_{k,i+1}^2 = \beta\sigma_{k,i}^2 + (\, 1 - \beta) \, |u_{k,i}|^2 \tag{B.54}$$

Update Step-Size:

$$A_k(i) = \alpha\mu_k(i) + \frac{\gamma}{L} \sum_{j=i-(\,L-1)}^{i} e_k^2(j)$$

$$\mu_k(i+1) = \begin{cases} A_k(i), & \text{if } i = kL \text{ and } A_k(i) \in (\, \mu_{k,min}, \mu_{k,max}) \\ \mu_{k,max}, & \text{if } i = kL \text{ and } A_k(i) \geq \mu_{k,max} \\ \mu_{k,min}, & \text{if } i = kL \text{ and } A_k(i) \leq \mu_{k,min} \\ \mu_k(i), & \text{if } i \neq kL \end{cases} \tag{B.55}$$

Update Clustering:

$$q_{k,i} = u_{k,i}^* (\, d_k(i) - u_{k,i}\psi_{k,i+1})$$

$$c_{kl}(i+1) = \frac{\| \, \psi_{k,i+1} + \mu_k(i+1) q_{k,i} - \psi_{l,i+1} \, \|^{-2}}{\sum_{j \in \mathcal{N}_k} \| \, \psi_{k,i+1} + \mu_k(i+1) q_{k,i} - \psi_{j,i+1} \, \|^{-2}} \tag{B.56}$$

Combine Step:

$$\phi_{k,i+1} = \sum_{l \in \mathcal{N}_k} c_{kl}(i+1)\psi_{l,i+1} \tag{B.57}$$

The theoretical derivation of transform domain variable step-size that considers the aforementioned equations for mean transient and mean-square analyses in order to achieve Multi-Task networks using TDVSS with adaptive combiners is short listed in Algorithm B.5 (B) for easy reference.

Algorithm B.5 (B): Theoretical MSD of ATC Diffusion LMS for Transform Domain Variable Step-Size DLMS (TDVSS) Multi-Task Networks with Adaptive Combiner.

Initialize: $C(0) = I_N$, $\eta(0) = \| \bar{w}^{(o)} \|^2$, $\bar{F}^0 = I_{N^2M^2}$, $\bar{F}_1^0 = I_{NM}$, $\eta_k(0) = \| \bar{w}^{(o)} \|^2$, $\zeta_k(0) = \bar{w}^o$ for all $k = 1, ..., N$, and $\bar{w}^o$ in transform domain.

$$\begin{aligned}\bar{F} = E\left[\bar{\boldsymbol{G}}^*_{i-1} \odot \bar{\boldsymbol{G}}^{*T}_{i-1}\right] [\, I_{N^2M^2} - (\, I_{NM} \odot \Lambda E\,[\boldsymbol{D}_{i-1}]) \\ - (\, \Lambda E\,[\boldsymbol{D}_{i-1}] \odot I_{NM}) + E\,[\boldsymbol{D}_{i-1} \odot \boldsymbol{D}_{i-1}]\,\mathcal{A}] \quad (N^2M^2 \times N^2M^2)\end{aligned} \tag{B.58}$$

$$\bar{F}_1 = [\, I_{NM} - E\,[\boldsymbol{D}_{i-1}]\,\Lambda]\, E\,[\boldsymbol{C}_{i-1}] \quad (NM \times NM) \tag{B.59}$$

Procedure: Start "for loop" at iteration $i \geq 1$ and node $1 \leq k \leq N$

Local node MSD:

$$\bar{\sigma}_k = \text{bvec}\{\mathbf{0}_{(k-1)M}, I_M, \mathbf{0}_{(N-k)M}\} \triangleq q_{k\eta} \quad (N^2M^2 \times 1) \tag{B.60}$$

$$\begin{aligned} \eta_k(i) &= \eta_k(i-1) + b^T \bar{F}^i q_{k\eta} - \parallel \bar{w}^{(o)} \parallel^2_{\bar{F}^i\left(I-\bar{F}\right)q_{k\eta}} \\ \eta_k^{'}(i) &= \eta_k(i) + b^T \bar{F}^i q_{k\eta} - \parallel \bar{w}^{(o)} \parallel^2_{\bar{F}^i\left(I-\bar{F}\right)q_{k\eta}} \end{aligned} \tag{B.61}$$

Local node mean:

$$q_{1,k\eta} = \text{bvec}\{\mathbf{0}_{(k-1)M}, 1_M, \mathbf{0}_{(N-k)M}\} \quad (NM \times 1) \tag{B.62}$$

$$\begin{aligned} \zeta_k(i) &= \zeta_k(i-1) - q_{1,k\eta}\left(I_N - \bar{F}_1\right)\bar{F}_1^i w^o \\ \zeta_k^{'}(i) &= \zeta_k(i) - q_{1,k\eta}\left(I_N - \bar{F}_1\right)\bar{F}_1^i w^o \end{aligned} \tag{B.63}$$

Update Clustering:

$$c_{kl}(i+1) = \frac{\left(\eta_k^{'}(i+1) + \eta_l(i+1) - 2\zeta_k^{'}(i+1)\zeta_l(i+1)\right)^{-1}}{\sum_{j \in \mathcal{N}_k}\left(\eta_k^{'}(i) + \eta_j(i+1) - 2\zeta_k^{'}(i+1)\zeta_j(i+1)\right)^{-1}} \tag{B.64}$$

Update Step-Size:

$$E\left[\boldsymbol{D}_i^2\right] = diag\{E\left[\boldsymbol{\mu}_1^{'}(i)\right]I_M, E\left[\boldsymbol{\mu}_2^{'}(i)\right]I_M, ..., E\left[\boldsymbol{\mu}_N^{'}(i)\right]I_M\}$$

$$E\left[\boldsymbol{D}_i\right] = diag\{E\left[\boldsymbol{\mu}_1^{'2}(i)\right] I_M, E\left[\boldsymbol{\mu}_2^{'2}(i)\right] I_M, ..., E\left[\boldsymbol{\mu}_N^{'2}(i)\right] I_M\} \tag{B.65}$$

$$\begin{aligned}
E\left[\boldsymbol{\mu}_k^{'}(i)\right] &= E\left[\boldsymbol{\mu}_k(i)\right]\left[\frac{1}{E\left[\boldsymbol{\sigma}_{k,i}^2\right]} - \frac{\delta}{E\left[\boldsymbol{\sigma}_{k,i}^4\right]}\right] \\
E\left[\boldsymbol{\mu}_k^{'2}(i)\right] &= E\left[\boldsymbol{\mu}_k^2(i)\right]\left[\frac{1}{E\left[\boldsymbol{\sigma}_{k,i}^4\right]} + \frac{\delta^2}{E\left[\boldsymbol{\sigma}_{k,i}^8\right]} - \frac{2\delta}{E\left[\boldsymbol{\sigma}_{k,i}^6\right]}\right]
\end{aligned} \tag{B.66}$$

$$\begin{aligned}
E\left[\boldsymbol{\mu}_k(i)\right] &= \alpha E\left[\boldsymbol{\mu}_k(i-1)\right] + \frac{\gamma}{L}\sum_{j=i-(L-1)}^{i} E\left[\boldsymbol{e}_k^2(j)\right] \\
E\left[\boldsymbol{\mu}_k^2(i)\right] &= \alpha^2 E\left[\boldsymbol{\mu}_k^2(i-1)\right] \\
&\quad + 2\alpha\frac{\gamma}{L}\sum_{j=i-(L-1)}^{i} E\left[\boldsymbol{e}_k^2(j)\right] E\left[\boldsymbol{\mu}_k(i-1)\right] \\
&\quad + \frac{\gamma^2}{L^2}\sum_{j=i-(L-1)}^{i} E\left[\boldsymbol{e}_k^4(j)\right]
\end{aligned} \tag{B.67}$$

$$\begin{aligned}
E\left[\boldsymbol{\sigma}_{k,i}^2\right] &= \beta E\left[\boldsymbol{\sigma}_{k,i-1}^2\right] + (1-\beta)\,\sigma_k^2 \\
E\left[\boldsymbol{\sigma}_{k,i}^4\right] &= \beta^2 E\left[\boldsymbol{\sigma}_{k,i-1}^4\right] + 2\beta(1-\beta)\,\sigma_k^2 E\left[\boldsymbol{\sigma}_{k,i-1}^2\right] + (1-\beta)^2\sigma_k^4 \\
E\left[\boldsymbol{\sigma}_{k,i}^6\right] &= \beta^3 E\left[\boldsymbol{\sigma}_{k,i-1}^6\right] + 3\beta^2(1-\beta)\,\sigma_k^2 E\left[\boldsymbol{\sigma}_{k,i-1}^4\right] + 3\beta(1-\beta)^2\sigma_k^4 E\left[\boldsymbol{\sigma}_{k,i-1}^2\right] \\
&\quad + (1-\beta)^3\sigma_k^6 \\
E\left[\boldsymbol{\sigma}_{k,i}^8\right] &= \beta^4 E\left[\boldsymbol{\sigma}_{k,i-1}^8\right] + 4\beta^3(1-\beta)\,\sigma_k^2 E\left[\boldsymbol{\sigma}_{k,i-1}^6\right] + 6\beta^2(1-\beta)^2\sigma_k^4 E\left[\boldsymbol{\sigma}_{k,i-1}^4\right] \\
&\quad + 4\beta(1-\beta)^3\sigma_k^6 E\left[\boldsymbol{\sigma}_{k,i-1}^2\right] + (1-\beta)^4\sigma_k^8
\end{aligned} \tag{B.68}$$

Global MSD:

$$\eta(i) = \eta(i-1) + b^T \bar{F}^i q_\eta - \parallel \bar{w}^{(o)} \parallel^2_{\bar{F}^i \left(I-\bar{F}\right) q_\eta} \tag{B.69}$$

B.6 New Variable Step-Size Transform-Domain DLMS (NVSTDLMS) with Adaptive Combiners

The variable step-size DLMS based on NVSTDLMS procedure for Multi-Task Networks with adaptive combiners is summarized in Algorithm B.6 (A) for easy simulation purposes.

Algorithm B.6 (A): ATC Diffusion LMS for New Variable Step-Size Transform-Domain DLMS (NVSTDLMS) Multi-Task Networks with Adaptive Combiner

---

Initialize: $C(0) = I_N$, $\phi_{k,0} = 0$ for all $k = 1, ..., N$, and $u_{k,i}$ in transform domain.

Procedure: Start "for loop" at iteration $i \geq 1$ and node $1 \leq k \leq N$

Adapt Step:

$$\psi_{k,i+1} = \phi_{k,i} + \frac{\mu_k(i)}{\delta + \sigma^2_{k,i}} u^*_{k,i}(\, d_k(i) - u_{k,i}\phi_{k,i}) \tag{B.70}$$

Update Power Estimate:

$$\sigma^2_{k,i+1} = \beta\sigma^2_{k,i} + (\,1-\beta)\,|u_{k,i}|^2 \tag{B.71}$$

Update Step-Size:

$$\mu_k(i+1) = \begin{cases} \frac{\varepsilon_{1,k}(i)}{\varepsilon_{2,k}(i)}, & \text{if } \mu_k(i+1) \in (\,\mu_{k,min}, \mu_{k,max}) \\ \mu_{k,max}, & \text{if } \mu_k(i+1) \geq \mu_{k,max} \\ \mu_{k,min}, & \text{if } \mu_k(i+1) \leq \mu_{k,min} \end{cases} \tag{B.72}$$

Update Clustering:

$$\begin{aligned} q_{k,i} &= u^*_{k,i}(\,d_k(i) - u_{k,i}\psi_{k,i+1}) \\ c_{kl}(i+1) &= \frac{\|\,\psi_{k,i+1} + \mu_k(i+1)q_{k,i} - \psi_{l,i+1}\,\|^{-2}}{\sum_{j\in\mathcal{N}_k}\|\,\psi_{k,i+1} + \mu_k(i+1)q_{k,i} - \psi_{j,i+1}\,\|^{-2}} \end{aligned} \tag{B.73}$$

Combine Step:

$$\phi_{k,i+1} = \sum_{l\in\mathcal{N}_k} c_{kl}(i+1)\psi_{l,i+1} \tag{B.74}$$

The theoretical derivation of variable step-size based on NVSTDLMS with adaptive networks that considers the aforementioned equations for mean transient and mean-square analyses in order to achieve Multi-Task networks is short listed in Algorithm B.6 (B) for easy reference.

Algorithm B.6 (B): Theoretical MSD of ATC Diffusion LMS for New Variable Step-Size Transform-Domain DLMS (NVSTDLMS) Multi-Task Networks with Adaptive Combiner.

Initialize: $C(0) = I_N$, $\eta(0) = \| \bar{w}^{(o)} \|^2$, $\bar{F}^0 = I_{N^2M^2}$, $\bar{F}_1^0 = I_{NM}$, $\eta_k(0) = \| \bar{w}^{(o)} \|^2$, $\zeta_k(0) = \bar{w}^o$ for all $k = 1, ..., N$, and $\bar{w}^o$ in transform domain.

$$\bar{F} = E\left[\bar{\boldsymbol{G}}_{i-1}^{*} \odot \bar{\boldsymbol{G}}_{i-1}^{*T}\right] [\, I_{N^2M^2} - (\, I_{NM} \odot \Lambda E\,[\boldsymbol{D}_{i-1}]) \\ - (\, \Lambda E\,[\boldsymbol{D}_{i-1}] \odot I_{NM}) + E\,[\boldsymbol{D}_{i-1} \odot \boldsymbol{D}_{i-1}]\, \mathcal{A}] \quad (N^2M^2 \times N^2M^2) \qquad \text{(B.75)}$$

$$\bar{F}_1 = [\, I_{NM} - E\,[\boldsymbol{D}_{i-1}]\, \Lambda]\, E\,[\boldsymbol{C}_{i-1}] \quad (NM \times NM) \qquad \text{(B.76)}$$

Procedure: Start "for loop" at iteration $i \geq 1$ and node $1 \leq k \leq N$

Local node MSD:

$$\bar{\sigma}_k = \text{bvec}\{\mathbf{0}_{(k-1)M}, I_M, \mathbf{0}_{(N-k)M}\} \triangleq q_{k\eta} \quad (N^2M^2 \times 1) \qquad \text{(B.77)}$$

$$\eta_k(i) = \eta_k(i-1) + b^T \bar{F}^i q_{k\eta} - \| \bar{w}^{(o)} \|^2_{\bar{F}^i \left(I - \bar{F}\right) q_{k\eta}} \\ \eta_k^{'}(i) = \eta_k(i) + b^T \bar{F}^i q_{k\eta} - \| \bar{w}^{(o)} \|^2_{\bar{F}^i \left(I - \bar{F}\right) q_{k\eta}} \qquad \text{(B.78)}$$

Local node mean:

$$q_{1,k\eta} = \text{bvec}\{\mathbf{0}_{(k-1)M}, 1_M, \mathbf{0}_{(N-k)M}\} \quad (NM \times 1) \tag{B.79}$$

$$\begin{aligned}
\zeta_k(i) &= \zeta_k(i-1) - q_{1,k\eta}\left(I_N - \bar{F}_1\right)\bar{F}_1^i w^o \\
\zeta_k^{'}(i) &= \zeta_k(i) - q_{1,k\eta}\left(I_N - \bar{F}_1\right)\bar{F}_1^i w^o
\end{aligned} \tag{B.80}$$

Update Clustering:

$$c_{kl}(i+1) = \frac{\left(\eta_k^{'}(i+1) + \eta_l(i+1) - 2\zeta_k^{'}(i+1)\zeta_l(i+1)\right)^{-1}}{\sum_{j\in\mathcal{N}_k}\left(\eta_k^{'}(i) + \eta_j(i+1) - 2\zeta_k^{'}(i+1)\zeta_j(i+1)\right)^{-1}} \tag{B.81}$$

Update Step-Size:

$$\begin{aligned}
E\left[\boldsymbol{D}_i^2\right] &= diag\{E\left[\boldsymbol{\mu}_1^{'}(i)\right]I_M, E\left[\boldsymbol{\mu}_2^{'}(i)\right]I_M, ..., E\left[\boldsymbol{\mu}_N^{'}(i)\right]I_M\} \\
E\left[\boldsymbol{D}_i\right] &= diag\{E\left[\boldsymbol{\mu}_1^{'2}(i)\right]I_M, E\left[\boldsymbol{\mu}_2^{'2}(i)\right]I_M, ..., E\left[\boldsymbol{\mu}_N^{'2}(i)\right]I_M\}
\end{aligned} \tag{B.82}$$

$$\begin{aligned}
E\left[\boldsymbol{\mu}_k^{'}(i)\right] &= E\left[\boldsymbol{\mu}_k(i)\right]\left[\frac{1}{E\left[\boldsymbol{\sigma}_{k,i}^2\right]} - \frac{\delta}{E\left[\boldsymbol{\sigma}_{k,i}^4\right]}\right] \\
E\left[\boldsymbol{\mu}_k^{'2}(i)\right] &= E\left[\boldsymbol{\mu}_k^2(i)\right]\left[\frac{1}{E\left[\boldsymbol{\sigma}_{k,i}^4\right]} + \frac{\delta^2}{E\left[\boldsymbol{\sigma}_{k,i}^8\right]} - \frac{2\delta}{E\left[\boldsymbol{\sigma}_{k,i}^6\right]}\right]
\end{aligned} \tag{B.83}$$

$$E\left[\boldsymbol{\mu}_k(i)\right] = \frac{\alpha_1 E\left[\boldsymbol{\varepsilon}_{1,k}(i-1)\right] + E\left[\boldsymbol{e}_k^2(i)\right]}{\alpha_2 E\left[\boldsymbol{\varepsilon}_{2,k}(i-1)\right] + E\left[\boldsymbol{e}_k^2(i)\right]}$$

$$E\left[\boldsymbol{\mu}_k^2(i)\right] = \frac{\alpha_1^2 E\left[\boldsymbol{\varepsilon}_{1,k}^2(i-1)\right] + 2\alpha_1 E\left[\boldsymbol{\varepsilon}_{1,k}(i-1)\right] E\left[\boldsymbol{e}_k^2(i)\right] + E\left[\boldsymbol{e}_k^4(i)\right]}{\alpha_2^2 E\left[\boldsymbol{\varepsilon}_{2,k}^2(i-1)\right] + 2\alpha_2 E\left[\boldsymbol{\varepsilon}_{2,k}(i-1)\right] E\left[\boldsymbol{e}_k^2(i)\right] + E\left[\boldsymbol{e}_k^4(i)\right]} \tag{B.84}$$

$$\begin{aligned}
E\left[\boldsymbol{\sigma}_{k,i}^2\right] &= \beta E\left[\boldsymbol{\sigma}_{k,i-1}^2\right] + (1-\beta)\,\sigma_k^2 \\
E\left[\boldsymbol{\sigma}_{k,i}^4\right] &= \beta^2 E\left[\boldsymbol{\sigma}_{k,i-1}^4\right] + 2\beta\,(1-\beta)\,\sigma_k^2 E\left[\boldsymbol{\sigma}_{k,i-1}^2\right] + (1-\beta)^2\,\sigma_k^4 \\
E\left[\boldsymbol{\sigma}_{k,i}^6\right] &= \beta^3 E\left[\boldsymbol{\sigma}_{k,i-1}^6\right] + 3\beta^2\,(1-\beta)\,\sigma_k^2 E\left[\boldsymbol{\sigma}_{k,i-1}^4\right] + 3\beta\,(1-\beta)^2\,\sigma_k^4 E\left[\boldsymbol{\sigma}_{k,i-1}^2\right] \\
&\quad + (1-\beta)^3\,\sigma_k^6 \\
E\left[\boldsymbol{\sigma}_{k,i}^8\right] &= \beta^4 E\left[\boldsymbol{\sigma}_{k,i-1}^8\right] + 4\beta^3\,(1-\beta)\,\sigma_k^2 E\left[\boldsymbol{\sigma}_{k,i-1}^6\right] + 6\beta^2\,(1-\beta)^2\,\sigma_k^4 E\left[\boldsymbol{\sigma}_{k,i-1}^4\right] \\
&\quad + 4\beta\,(1-\beta)^3\,\sigma_k^6 E\left[\boldsymbol{\sigma}_{k,i-1}^2\right] + (1-\beta)^4\,\sigma_k^8
\end{aligned} \tag{B.85}$$

Global MSD:

$$\eta(i) = \eta(i-1) + b^T \bar{F}^i q_\eta - \parallel \bar{w}^{(o)} \parallel^2_{\bar{F}^i\left(I-\bar{F}\right)q_\eta} \tag{B.86}$$

B.7 Optimal Variable Step-Size Transform-Domain DLMS (VSSTDLMS) with Adaptive Combiners

The optimal variable step-size transform domain DLMS (VSSTDLMS) procedure for Multi-Task Networks with adaptive combiners is summarized in Algorithm B.7 (A) for easy simulation purposes.

Algorithm B.7 (A): ATC Diffusion LMS for optimal Variable Step-Size Transform Domain DLMS (VSSTDLMS) Multi-Task Networks Algorithm with Adaptive Combiner

---

Initialize: $C(0) = I_N$, $\phi_{k,0} = 0$ for all $k = 1, ..., N$, and $u_{k,i}$ in transform domain.

Procedure: Start "for loop" at iteration $i \geq 1$ and node $1 \leq k \leq N$

Adapt Step:

$$\psi_{k,i+1} = \phi_{k,i} + \frac{\mu_k(i)}{\delta + \sigma_{k,i}^2} u_{k,i}^* (d_k(i) - u_{k,i}\phi_{k,i}) \tag{B.87}$$

Update Power Estimate:

$$\sigma_{k,i+1}^2 = \beta\sigma_{k,i}^2 + (1 - \beta) |u_{k,i}|^2 \tag{B.88}$$

Update Step-Size:

$$\rho_k(i+1) = \gamma\rho_k(i) + (1-\gamma)\ \left(e_k^2(i)\hat{u}_{k,i}^T\hat{u}_{k,i}\right)$$
$$\eta_k(i+1) = \gamma\eta_k(i) + (1-\gamma)\ \big(e_k(i)e_k(i-1)\big)$$
$$\mu_k(i+1) = \frac{\big|\eta_k(i+1)\big|}{\rho_k(i+1)} \tag{B.89}$$

Update Clustering:

$$q_{k,i} = u_{k,i}^*(\,d_k(i) - u_{k,i}\psi_{k,i+1})$$
$$c_{kl}(i+1) = \frac{\|\ \psi_{k,i+1} + \mu_k q_{k,i} - \psi_{l,i+1}\ \|^{-2}}{\sum_{j\in\mathcal{N}_k}\|\ \psi_{k,i+1} + \mu_k q_{k,i} - \psi_{j,i+1}\ \|^{-2}} \tag{B.90}$$

Combine Step:

$$\phi_{k,i+1} = \sum_{l\in\mathcal{N}_k} c_{kl}(i+1)\psi_{l,i+1} \tag{B.91}$$

The theoretical derivation of optimal variable step-size that considers the aforementioned equations for mean transient and mean-square analyses in order to achieve Multi-Task Networks with adaptive combiners is short listed in Algorithm B.7 (B) for easy reference.

Algorithm B.7 (B): Theoretical MSD of ATC Diffusion LMS for optimal variable step-size transform domain DLMS (VSSTDLMS) Multi-Task Networks with Adaptive Combiner.

Initialize: $C(0) = I_N$, $\eta(0) = \| \bar{w}^{(o)} \|^2$, $\bar{F}^0 = I_{N^2M^2}$, $\bar{F}_1^0 = I_{NM}$, $\eta_k(0) = \| \bar{w}^{(o)} \|^2$, $\zeta_k(0) = \bar{w}^o$ for all $k = 1, ..., N$, and $\bar{w}^o$ in transform domain.

$$\bar{F} = E\left[\bar{\boldsymbol{G}}_{i-1}^{*} \odot \bar{\boldsymbol{G}}_{i-1}^{*T}\right]\left[\, I_{N^2M^2} - (\, I_{NM} \odot \Lambda E\left[\boldsymbol{D}_{i-1}\right]) \right.$$
$$\left. - (\, \Lambda E\left[\boldsymbol{D}_{i-1}\right] \odot I_{NM}) + E\left[\boldsymbol{D}_{i-1} \odot \boldsymbol{D}_{i-1}\right] \mathcal{A}\right] \quad (N^2M^2 \times N^2M^2) \tag{B.92}$$

$$\bar{F}_1 = \left[\, I_{NM} - E\left[\boldsymbol{D}_{i-1}\right] \Lambda\right] E\left[\boldsymbol{C}_{i-1}\right] \quad (NM \times NM) \tag{B.93}$$

Procedure: Start "for loop" at iteration $i \geq 1$ and node $1 \leq k \leq N$

Local node MSD:

$$\bar{\sigma}_k = \text{bvec}\{\mathbf{0}_{(k-1)M}, I_M, \mathbf{0}_{(N-k)M}\} \triangleq q_{k\eta} \quad (N^2M^2 \times 1) \tag{B.94}$$

$$\eta_k(i) = \eta_k(i-1) + b^T \bar{F}^i q_{k\eta} - \| \bar{w}^{(o)} \|^2_{\bar{F}^i\left(I-\bar{F}\right)q_{k\eta}}$$
$$\eta_k^{'}(i) = \eta_k(i) + b^T \bar{F}^i q_{k\eta} - \| \bar{w}^{(o)} \|^2_{\bar{F}^i\left(I-\bar{F}\right)q_{k\eta}} \tag{B.95}$$

Local node mean:

$$q_{1,k\eta} = \text{bvec}\{\mathbf{0}_{(k-1)M}, 1_M, \mathbf{0}_{(N-k)M}\} \quad (NM \times 1) \tag{B.96}$$

$$\begin{aligned}
\zeta_k(i) &= \zeta_k(i-1) - q_{1,k\eta}\left(I_N - \bar{F}_1\right)\bar{F}_1^i w^o \\
\zeta_k^{'}(i) &= \zeta_k(i) - q_{1,k\eta}\left(I_N - \bar{F}_1\right)\bar{F}_1^i w^o
\end{aligned} \tag{B.97}$$

Update Clustering:

$$c_{kl}(i+1) = \frac{\left(\eta_k^{'}(i+1) + \eta_l(i+1) - 2\zeta_k^{'}(i+1)\zeta_l(i+1)\right)^{-1}}{\sum_{j\in\mathcal{N}_k}\left(\eta_k^{'}(i) + \eta_j(i+1) - 2\zeta_k^{'}(i+1)\zeta_j(i+1)\right)^{-1}} \tag{B.98}$$

Update Step-Size:

$$\begin{aligned}
E\left[\boldsymbol{D}_i^2\right] &= diag\{E\left[\boldsymbol{\mu}_1^{'}(i)\right]I_M, E\left[\boldsymbol{\mu}_2^{'}(i)\right]I_M, ..., E\left[\boldsymbol{\mu}_N^{'}(i)\right]I_M\} \\
E\left[\boldsymbol{D}_i\right] &= diag\{E\left[\boldsymbol{\mu}_1^{'2}(i)\right]I_M, E\left[\boldsymbol{\mu}_2^{'2}(i)\right]I_M, ..., E\left[\boldsymbol{\mu}_N^{'2}(i)\right]I_M\}
\end{aligned} \tag{B.99}$$

$$\begin{aligned}
E\left[\boldsymbol{\mu}_k^{'}(i)\right] &= E\left[\boldsymbol{\mu}_k(i)\right]\left[\frac{1}{E\left[\boldsymbol{\sigma}_{k,i}^2\right]} - \frac{\delta}{E\left[\boldsymbol{\sigma}_{k,i}^4\right]}\right] \\
E\left[\boldsymbol{\mu}_k^{'2}(i)\right] &= E\left[\boldsymbol{\mu}_k^2(i)\right]\left[\frac{1}{E\left[\boldsymbol{\sigma}_{k,i}^4\right]} + \frac{\delta^2}{E\left[\boldsymbol{\sigma}_{k,i}^8\right]} - \frac{2\delta}{E\left[\boldsymbol{\sigma}_{k,i}^6\right]}\right]
\end{aligned} \tag{B.100}$$

$$
\begin{aligned}
E\left[\boldsymbol{\sigma}_{k,i}^{2}\right] &= \beta E\left[\boldsymbol{\sigma}_{k,i-1}^{2}\right] + (1-\beta)\,\sigma_k^2 \\
E\left[\boldsymbol{\sigma}_{k,i}^{4}\right] &= \beta^2 E\left[\boldsymbol{\sigma}_{k,i-1}^{4}\right] + 2\beta\,(1-\beta)\,\sigma_k^2 E\left[\boldsymbol{\sigma}_{k,i-1}^{2}\right] + (1-\beta)^2\,\sigma_k^4 \\
E\left[\boldsymbol{\sigma}_{k,i}^{6}\right] &= \beta^3 E\left[\boldsymbol{\sigma}_{k,i-1}^{6}\right] + 3\beta^2\,(1-\beta)\,\sigma_k^2 E\left[\boldsymbol{\sigma}_{k,i-1}^{4}\right] + 3\beta\,(1-\beta)^2\,\sigma_k^4 E\left[\boldsymbol{\sigma}_{k,i-1}^{2}\right] \\
&\quad + (1-\beta)^3\,\sigma_k^6 \\
E\left[\boldsymbol{\sigma}_{k,i}^{8}\right] &= \beta^4 E\left[\boldsymbol{\sigma}_{k,i-1}^{8}\right] + 4\beta^3\,(1-\beta)\,\sigma_k^2 E\left[\boldsymbol{\sigma}_{k,i-1}^{6}\right] + 6\beta^2\,(1-\beta)^2\,\sigma_k^4 E\left[\boldsymbol{\sigma}_{k,i-1}^{4}\right] \\
&\quad + 4\beta\,(1-\beta)^3\,\sigma_k^6 E\left[\boldsymbol{\sigma}_{k,i-1}^{2}\right] + (1-\beta)^4\,\sigma_k^8 \qquad \text{(B.101)}
\end{aligned}
$$

Global MSD:

$$
\eta(i) = \eta(i-1) + b^T \bar{F}^i q_\eta - \parallel \bar{w}^{(o)} \parallel^2_{\bar{F}^i\left(I-\bar{F}\right)q_\eta} \qquad \text{(B.102)}
$$

Appendix-C Jensen's Inequality

A function can be convex, strict convex, or concave function. For instance, the convex functions include:

- x^2
- $|x|$
- e^x
- $x \log x \quad \text{and} \quad x \geq 0$

while concave functions include

- $\sqrt{x}$
- $\log x \quad \text{and} \quad x \geq 0$

C.1 Convex and Concave Functions

We begin with the properties of convex and concave functions [28].

Definition (Convex Space):

A space $\mathbb{S}$ is a set of object in which addition is defined as

$$x_1 + x_2 \in \mathbb{S} \tag{C.1}$$

for every x_1 and x_2 in $\mathbb{S}$.

The multiplication is defined as

$$ax \in \mathbb{S} \tag{C.2}$$

where a is a constant in $\mathbb{S}$.

Also, for a constant θ, we have

$$\theta x_1 + (1 - \theta)\, x_2 \in \mathbb{S} \tag{C.3}$$

for every x_1 and x_2 in $\mathbb{S}$ and $0 \leq \theta \leq 1$.

The convex function is called Cup function (U function) while the concave function is called Cap function.

Definition (Convex Vs. Concave Functions):

A real-valued function with domain $\mathbb{S}$ is called a convex function iff for any x_1, x_2 $\in \mathbb{S}$ and $0 \leq \theta \leq 1$,

$$f(\,\theta x_1 + (1 - \theta)\, x_2) \leq \theta f(x_1) + (1 - \theta)\, f(x_2) \tag{C.4}$$

and it is called concave function iff

$$f(\theta x_1 + (1-\theta)x_2) \geq \theta f(x_1) + (1-\theta)f(x_2) \tag{C.5}$$

Equality holds if it is a strictly convex function. If a function is both convex and concave functions, then it is linear.

C.2 Jensen's Inequality

Definition (Jensen's Inequality):

If a function f is convex (U) and defined on a convex space, then

$$E[f(x)] \leq f(E[x]) \tag{C.6}$$

On the other hand, if a function f is concave and defined on a concave space, then

$$E[f(x)] \geq f(E[x]) \tag{C.7}$$

For example, for two values x_1 and x_2 in $\mathbb{S}$, we have

$$p(x_1)f(x_1) + p(x_2)f(x_2) \leq f(x_1 p(x_1) + x_2 p(x_2)) \tag{C.8}$$

REFERENCES

[1] A. H. Sayed, *Fundamentals of Adaptive Filtering.* Hoboken, NJ: J. Wiley & Sons [New York], 2003. [Online]. Available: http://opac.inria.fr/record=b1100673

[2] J. Chen, C. Richard, and A. H. Sayed, "Multitask Diffusion Adaptation over Networks," *CoRR*, vol. abs/1311.4894, 2013. [Online]. Available: http://arxiv.org/abs/1311.4894

[3] R. H. Kwong and E. W. Johnston, "A Variable Step Size LMS Algorithm," *IEEE Transactions on Signal Processing*, vol. 40, no. 7, pp. 1633–1642, Jul 1992.

[4] C. G. Lopes and A. H. Sayed, "Diffusion Least-Mean Squares Over Adaptive Networks: Formulation and Performance Analysis," *Signal Processing, IEEE Transactions on*, vol. 56, no. 7, pp. 3122–3136, July 2008.

[5] S. Narayan, A. Peterson, and M. Narasimha, "Transform Domain LMS Algorithm," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 31, no. 3, pp. 609–615, Jun 1983.

[6] J. Lee and C. Un, "Performance of Transform-Domain LMS Adaptive Digital Filters," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 34, no. 3, pp. 499–510, Jun 1986.

[7] F. Beaufays, "Transform-Domain Adaptive Filters: An Analytical Approach," *Signal Processing, IEEE Transactions on*, vol. 43, no. 2, pp. 422–431, Feb 1995.

[8] D. I. Kim and P. D. Wilde, "Performance Analysis of the DCT-LMS Adaptive Filtering Algorithm," *Signal Processing*, vol. 80, no. 8, pp. 1629–1654, 2000.

[9] R. C. Bilcu, P. Kuosmanen, and K. Egiazarian, "A Transform Domain LMS Adaptive Filter with Variable Step-Size," *Signal Processing Letters, IEEE*, vol. 9, no. 2, pp. 51–53, Feb 2002.

[10] K. Mayyas, "A Transform Domain LMS Algorithm with an Adaptive Step Size Equation," in *Signal Processing and Information Technology, 2004. Proceedings of the Fourth IEEE International Symposium on*, Dec 2004, pp. 229–232.

[11] Z. Shengkui, M. Zhihong, and K. Suiyang, "A New Variable Step-Size Transform Domain LMS Algorithm with System Identification," in *Control and Automation, 2007. ICCA 2007. IEEE International Conference on*, May 2007, pp. 2721–2724.

[12] D. I. Kim and P. Wilde, "Performance Analysis of The DCT-LMS Adaptive Filtering Algorithm," *Signal Processing*,

vol. 80, no. 8, pp. 1629 – 1654, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165168400000980

[13] G. Ying and X. Shengli, "A New Variable Step-Size LMS Adaptive Filtering Algorithm and its Performance Analysis," in *Acta Electronica Sinica*, vol. 29, no. 8, 2001, pp. 1094–1097.

[14] M. O. Bin Saeed, A. Zerguine, and S. A. Zummo, "A Variable Step-Size Strategy For Distributed Estimation Over Adaptive Networks," *EURASIP Journal on Advances in Signal Processing*, vol. 2013, no. 1, p. 135, Aug 2013. [Online]. Available: https://doi.org/10.1186/1687-6180-2013-135

[15] J. J. Shynk, "Frequency-domain and multirate adaptive filtering," *IEEE Signal Processing Magazine*, vol. 9, no. 1, pp. 14–37, Jan 1992.

[16] C. G. Lopes and A. H. Sayed, "Incremental Adaptive Strategies Over Distributed Networks," *Signal Processing, IEEE Transactions on*, vol. 55, no. 8, pp. 4064–4077, Aug 2007.

[17] J. Chen, C. Richard, and A. H. Sayed, "Diffusion LMS Over Multitask Networks," *IEEE Transactions on Signal Processing*, vol. 63, no. 11, pp. 2733–2748, June 2015.

[18] N. Takahashi, I. Yamada, and A. H. Sayed, "Diffusion Least-Mean Squares With Adaptive Combiners: Formulation and Performance Analysis," *IEEE Transactions on Signal Processing*, vol. 58, no. 9, pp. 4795–4810, Sept 2010.

[19] M. Wahab, M. Uzzaman, M. Hai, M. Haque, and M. Hasan, "Least-Squares Optimal Variable Step-Size LMS for Nonblind System Identification with Noise," in *Electrical and Computer Engineering, 2008. ICECE 2008. International Conference on*, Dec 2008, pp. 428–433.

[20] T. Jingfan and W. Gang, "The Variable Step Size LMS Adaptive Filtering Algorithm Based on S function," in *Radio engineering*, 1996, pp. 44–47.

[21] B. Lu, C. Feng, and G. Long, "A New Variable Step-Size LMS Adaptive Algorithm Based on Marr Function," in *Information Technology and Applications (ITA), 2013 International Conference on*, Nov 2013, pp. 214–217.

[22] G. Murmu and R. Nath, "Convergence Performance Comparison of Transform Domain LMS Adaptive Filters for Correlated Signal," in *Devices and Communications (ICDeCom), 2011 International Conference on*, Feb 2011, pp. 1–5.

[23] S. Attallah, "Wavelet Transform Domain Adaptive FIR Filtering," *CIRCUITS AND SYSTEMS II: ANALOG AND DIGITAL SIGNAL PROCESSING, IEEE Transactions on*, vol. 47, no. 3, pp. 209–213, Mar 2000.

[24] G. Murmu and S. Bhattacharya, "A Comparative Study of Different Wavelets Used in Transformed Domain LMS Filter," in *India Conference (INDICON), 2011 Annual IEEE*, Dec 2011, pp. 1–4.

[25] S. A. Zummo, B. S. M. Omer, and Z. Azzedine, "Variable Step-Size Least Mean Square Method for Estimation in Adap-

tive Networks," Patent 20 120 106 357, May, 2012. [Online]. Available: http://www.freepatentsonline.com/y2012/0106357.html

[26] A. Zerguine, M. Saeed, and S. Zummo, "Noise-Constrained Diffusion Least Mean Square Method for Estimation in Adaptive Networks," Patent, jun, 2013, uS Patent 8,462,892. [Online]. Available: http://www.google.com/patents/US8462892

[27] M. O. Bin Saeed, A. Zerguine, and S. A. Zummo, "Variable Step-Size Least Mean Square Algorithms Over Adaptive Networks," in *Information Sciences Signal Processing and their Applications (ISSPA), 2010 10th International Conference on*, May 2010, pp. 381–384.

[28] T. M. Cover and J. A. Thomas, *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing).* Wiley-Interscience, 2006.

Vitae

- Name: Ali Mohammed Al-Mohammedi
- Nationality: Kingdom of Saudi Arabia
- Date of Birth: March 5th, 1977 in Khobar, KSA
- Email: *G201004160@kfupm.edu.sa*
- Permanent Address: 9 Saleh Al-Sultan Street, P.O. Box 122 Dammam 31411 Kingdom of Saudi Arabia
- B.Sc, Electrical Engineering, Purdue University, IN, USA, May 2003
- M.Sc, Electrical and Computer Engineering, Oakland University, Rochester, MI, USA, August 2005
- M.Sc, Electrical Engineering, Polytechnic Institute of New York University, Brooklyn, NY, USA, January 2010
- Certificate, Wireless Communications, Polytechnic Institute of New York University, Brooklyn, NY, USA, January 2011
- Certificate, Power Electronics and Systems, Polytechnic Institute of New York University, Brooklyn, NY, USA, January 2011
- M.Sc, Telecommunications Engineering, King Fahd University of Petroleum and Minerals, Dhahran, Eastern Province, KSA, December 2017

- Worked for Saudi Basic Industries Corporation (SABIC), Eastern Petrochemical Company (SHARQ), P.E. Maintenance, 3rd SHARQ Expansion project, and Reliability/Engineering Departments, Jubail Industrial City, KAS, between 1999-2002.

- Worked for Saudi Electricity Company (SEC), Distribution Department, Electrical Engineer, Phase-III Building in Head Quarter, Dammam, KAS, 2010

- Worked for Jacobs Engineering, Electrical Engineering Department, Khober, KSA, 2011-2013.

- Worked for HADI Company, Department Manager, Administrative Communication Department, Dammam, KSA, 2013-2014.

- Worked for Ministry of Foreign Affairs (Aldewan), Asset Department, Building Adminstration Section, Electrical Engineering, Riyadh, KSA, 2014-2015.

- Currently, working for Dammam Airports Company (DACO), previously known as King Fahd International Airport (KFIA), Utilities and Equipment Department, Supervisor of Utilities Maintenance, Dammam, KSA, since 2015